ESTIMATING SINGLE AND MULTIPLE TARGET LOCATIONS USING K-MEANS CLUSTERING WITH RADIO TOMOGRAPHIC IMAGING IN WIRELESS SENSOR NETWORKS

THESIS

Jeffrey K. Nishida, Captain, USAF

AFIT-ENG-MS-15-M-038

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

AFIT-ENG-MS-15-M-038

ESTIMATING SINGLE AND MULTIPLE TARGET LOCATIONS USING K-MEANS CLUSTERING WITH RADIO TOMOGRAPHIC IMAGING IN WIRELESS SENSOR NETWORKS

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Jeffrey K. Nishida, B.S.E.E.
Captain, USAF

March 2015

AFIT-ENG-MS-15-M-038

ESTIMATING SINGLE AND MULTIPLE TARGET LOCATIONS USING K-MEANS CLUSTERING WITH RADIO TOMOGRAPHIC IMAGING IN WIRELESS SENSOR NETWORKS

Jeffrey K. Nishida, B.S.E.E.
Captain, USAF

Committee Membership:

Richard K. Martin, PhD
Chair

Captain Jesse D. Peterson, PhD
Member

Jason R. Pennington, PhD
Member

AFIT-ENG-MS-15-M-038

Abstract

Geolocation involves using data from a sensor network to assess and estimate the location of a moving or stationary target. Received Signal Strength (RSS), Angle of Arrival (AoA), and/or Time Difference of Arrival (TDoA) measurements can be used to estimate target location in sensor networks. Radio Tomographic Imaging (RTI) is an emerging Device-Free Localization (DFL) concept that utilizes the RSS values of a Wireless Sensor Network (WSN) to geolocate stationary or moving target(s). The WSN is set up around the Area of Interest (AoI) and the target of interest, which can be a person or object. The target inside the AoI creates a shadowing loss between each link being obstructed by the target. This research focuses on position estimation of single and multiple targets inside a RTI network. This research applies K-means clustering to localize one or more targets. K-means clustering is an algorithm that has been used in data mining applications such as machine learning applications, pattern recognition, hyper-spectral imagery, artificial intelligence, crowd analysis, and Multiple Target Tracking (MTT).

Acknowledgments

I would like to thank my wife for her love and support through this graduate program. I am also thankful for the friendships my wife and I made. Their company, encouragement, and humor have made this assignment enjoyable. I would also like to thank my advisor, Dr. Richard Martin for his guidance and sharing his expertise. Lastly, thank you to the sponsors, AFOSR.

Jeffrey K. Nishida

Table of Contents

List of Figures

Figure Page

List of Tables

List of Symbols

Symbol	Definition
$\mathbf{0}$	Vector or Matrix of Zeros
A	Area (ft^2)
α	Tunable Regularization Parameter
β	Tunable *Total Variation Regularization* Sharpness Parameter
c_H	Center Cartesian Coordinates of Human Cylindrical Model (ft)
$\mathbf{C}$	Covariance Matrix $[M \times N]$
d	Distance (ft)
$\mathbf{D}$	Difference Operator Matrix $[M \times N]$
δ_c	Pixel Correlation Constant (ft)
Δ	Change or Difference
Δ_p	Pixel Side Length (ft)
$\boldsymbol{\Lambda}$	Diagonal Matrix of Singular Values (Non-negative Real Numbers) $[M \times N]$
ϵ_D	*Euclidean Distance* Position Estimate Error (ft)
η_p	Path Loss Parameter
F	Fading Loss from Constructive and Destructive Interference of Narrow-band Signals in a Multipath Environment (dB)
γ	Tunable CRLB Surface Parameter

Symbol	Definition
$\mathbf{I}$	Identity Vector or Matrix
J	Cost Function
$\mathbf{J}$	*Fisher Information Matrix*
k	Cluster Number
K	Number of Clusters
l	Link Counter
L	Static Loss Due to Distance, Antenna Patterns, Device Inconsistencies, etc... (dB)
Lx, Ly, Lz	Number of Pixels or Voxels in Respective Cartesian Dimension
λ	Width of Weighting Ellipse (ft)
λ_i	Diagonal Entries of $\mathbf{\Lambda}$ (Square Root of Non-zero Eigenvalue)
M	Number of Two-Way Links
μ	*Ledoit-Wolf Estimator* Identity Matrix Scaling Parameter
n	Image Frame
$\mathbf{n}$	Noise (dB) $[M \times 1]$
N	Number of Pixels
∇	Gradient
ν	*Ledoit-Wolf Estimator* Covariance Matrix Shrinkage Parameter
$\mathbf{\Omega}$	Signal Attenuation Weight Assigned to Pixel or Voxel (scalar)

Symbol	Definition
p	Pixel or Counter
P	Signal Power (dB)
$p_X(x)$	Probability Density/Mass Function
$p_Y(y\|x)$	Conditional Probability Density/Mass Function
Π	Linear Transformation Matrix
Π_0	Power Loss (dB) at a Short Reference Distance, d_0 (ft)
$\mathbf{Q}$	*Tikhonov Regularization* Matrix Operator $[M \times N]$
r	RSS Measurement (dB)
R_H	Radius for Human Cylindrical Model (ft)
R_c	Cluster Radius for K-means Localization
$\mathbf{S}$	Affected Pixel or Voxel Binary Selection Matrix $[M \times N]$
S	Shadowing Loss Due to Objects that Attenuate the Signal (dB)
σ^2	Variance (dB^2)
σ	Standard Deviation (dB)
$\sum$	Sum
t	Time
u_i	Orthonormal Vector Entries of the Unitary Matrix $\mathbf{U}$ (Eigenvector Corresponding to λ_i)
$\mathbf{U}$	Unitary Matrix $[M \times M]$

Symbol	Definition
$\hat{\mathbf{U}}$	Intrinsic Subspace
$\tilde{\mathbf{U}}$	Extrinsic Subspace
v	Measurement Noise (dB)
$\mathbf{V}$	Unitary Matrix $[N \times N]$
$\mathbf{W}$	Transfer Matrix of Pixel Signal Attenuation Weight Model Parameters, $\mathbf{\Omega} \cdot \mathbf{S}$ $[M \times N]$
x	Pixel Signal Attenuation (dB/ft)
$\mathbf{x}$	Collection of Pixel Signal Attenuation (dB/ft) $[N \times 1]$
$\mathbf{y}$	Collection of Calibrated RSS Link Measurements (dB) $[M \times 1]$
$\hat{\mathbf{x}}$	Attenuation Image Estimate (dB/ft)
$\hat{\mathbf{y}}$	Intrinsic Signal Component
$\tilde{\mathbf{y}}$	Extrinsic Signal Component
Z	Total Static Fading Loss (dB)
$\sim$	Distributed As
(x, y)	2-D Cartesian Coordinates
(x, y, z)	3-D Cartesian Coordinates
$(\bar{\bullet})$	Ensemble or Sample Mean of the Argument
$(\hat{\bullet})$	Estimate of the Argument
$\|\| \bullet \|\|_{l_1}$	l_1 Norm of the Argument

Symbol	Definition
$\|\| \bullet \|\|_{l_2}$	l_2 Norm or *Euclidean Distance* of the Argument (ft)
$\|\| \bullet \|\|_{weight}$	Weighted Least Squares of the Argument
$(\bullet)^{-1}$	Matrix or Vector Inverse of the Argument
$(\bullet)^{T}$	Matrix or Vector Transpose of the Argument
$f(\bullet)$	Objective Cost Function of the Argument
$Var(\bullet)$	Variance of the Argument
$Cov(\bullet)$	Covariance of the Argument
$\mathcal{N}(\bullet, \bullet)$	*Gaussian Distribution*

Symbol	Definition
	Subscripts
c	Calibration
$chan$	Channel
CHM	Cylindrical Human Model
D	Distance
E	Extrinsic
$fade$	Fade-Level
H	Human
i	Receiving Node
I	Intrinsic
j	Transmitting Node
k	K-means Cluster
l	Link
l_1	l_1 Norm
l_2	Euclidean Norm
M	Number of Unique Two-way Links
n	Noise or Image Frame
N	Number of Wireless Nodes
O	Obstructed
p	Pixel
r	Real-time RSS Measurement
reg	Regularized Term
R	RSS or Real-Time
RLS	Regularized Least Squares
SUB	SubVRT Estimation

Symbol	Definition
t	time
T	Transmit
TIK	Tikhonov Regularization
WLS	Weighted Least Squares
x	Pixel or Voxel Signal Attenuation
X	Transverse Horizontal Direction (width)
Y	Horizontal Direction (depth)
Z	Vertical Direction (height)

Superscripts

T	matrix transpose
-1	matrix inverse
δ	+ or - Sign of Change

List of Acronyms

Acronym	Definition
2-D	two-dimensional
3-D	three-dimensional
AFIT	Air Force Institute of Technology
AFRL	Air Force Research Laboratory
AoA	Angle of Arrival
AoI	Area of Interest
AWGN	Additive White Gaussian Noise
CRLB	Cramer-Rao Lower Bound
dB	decibels
DFL	Device-Free Localization
DRTI	Direction-based Radio Tomographic Imaging
GPS	Global Positioning System
HRTI	Histogram-based Radio Tomographic Imaging
GUI	Graphical User Interface
ICD	Informed Consent Document
Inc.	Incorporated
IR	Infrared
IRB	Institutional Review Board
KRTI	Kernel-based Radio Tomographic Imaging
LOS	Line-of-Sight
MAP	Maximum A-posteriori Probability
MEMS	Micro Electro-Mechanical Systems
MIMO	Multiple Input, Multiple Output

Acronym	Definition
MRTI	Mean-based Radio Tomographic Imaging
MSE	Mean Squared Error
MTT	Multiple Target Tracking
NaN	Not a Number
OS	Operating System
PVC	Polyvinyl Chloride
RF	Radio Frequency
RFIC	Radio Frequency Integrated Circuit
RMSE	Root Mean Squared Error
RTI	Radio Tomographic Imaging
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SOCHE	Southwestern Ohio Council for Higher Education
SPAN	Sensing and Processing Across Networks
TDoA	Time Difference of Arrival
TSVD	Truncated Singular Value Decomposition
TV	Total Variation
UC	University of California
USB	Universal Serial Bus
UWB	Ultra-Wideband
VRTI	Variance-based Radio Tomographic Imaging
WLAN	Wireless Local Area Networks
WSN	Wireless Sensor Network

ESTIMATING SINGLE AND MULTIPLE TARGET LOCATIONS USING K-MEANS CLUSTERING WITH RADIO TOMOGRAPHIC IMAGING IN WIRELESS SENSOR NETWORKS

I. Introduction

This chapter provides background on the methods and application of WSNs and Radio Tomographic Imaging (RTI). The thesis problem statement, assumptions, research objectives, approach used, and structure for this thesis are contained in this chapter.

1.1 Background

The growth and maturity of wireless communication and Micro Electro-Mechanical Systems (MEMS) technology has laid the foundation for the use of low power, low cost Radio Frequency (RF) sensors in various geolocation tasks [4]. A WSN involves multiple Radio Frequency Integrated Circuits (RFICs) deployed around an area of interest. The RFICs are often referred to as a radio, node, or mote which can be used interchangeably. Each node in the network is capable of sending and receiving information over a wireless communication channel. A variety of applications have been explored to utilize the use of WSN to support both military and civilian applications. Although geolocation with Ultra-Wideband (UWB) radar has provided much of the framework in WSN applications [5], [6], WSN differ such that a larger amount of nodes can be deployed. This is feasible because such networks are mobile, have flexible uses, and are easily implemented due to their low cost. WSNs with a large amount of nodes have uses in inventory monitoring, surveillance, classification, and localization [7].

Recent research into the application and effectiveness of WSNs for the use in surveillance, localization, and classification have led to an interest from military, special forces, and the emergency response community [1], [8], [9], [10].

1.2 Radio Tomographic Imaging

Geolocation involves using data from a sensor network to asses and estimate the location of a moving or stationary target. RSS, AoA, and/or TDoA measurements can be used to estimate target location in sensor networks. RTI uses the RSS information from each radio to estimate the position of the target(s). RTI is an emerging DFL concept that utilizes the RSS values of a WSN to geolocate a stationary or moving target. Every wireless node is a 2-way communication link that can transmit and receive RSS values over the specified communication channel [5]. The WSN is set up around the AoI and the target of interest, which can be a person or object. The target inside the AoI creates a shadowing loss between each link being obstructed by the target. This research focuses on position estimation of single and multiple targets inside a RTI network. In the literature, the focus has been on single targets using a Maximum A-posteriori Probability (MAP) estimator [5], [1], [10]. This research will apply K-means clustering to localize one or more targets. K-means clustering is a known algorithm used in other data mining applications such as among machine learning applications, pattern recognition, hyper-spectral imagery, artificial intelligence, crowd analysis, and MTT [11], [12], [13].

1.3 Problem Statement

Can K-means clustering be utilized with an indoor RTI network to localize one or more targets?

The motivation behind using K-means clustering is to provide an alternative means to localize target(s) inside a RTI network. Additionally, localizing multiple targets in RTI has been a difficult task. MTT has possible law enforcement, special forces, and military application. For example, in applications such as a hostage situation, special forces would want to be able to localize multiple targets inside the building [10], [11], [12]. MTT would be useful in gaining insight to where all targets inside the AoI are located.

1.4 Approach

This thesis will include theoretical analysis and background as the foundation of this research. The use of simulation and physical experiments will be used to support the objective of this research. RTI experiments done in real-time and with obstructions provide experimental results that can be analyzed. The data from the WSN will be collected in the form of RSS measurements and the information from the network will illustrate the attenuation caused by the affects of targets inside the network. Regularization, weighting models, image reconstruction, and localization estimation techniques will be used to provide results to be compared with simulations.

1.5 Thesis Structure

The remainder of this research document is arranged into four chapters. Chapter 2 provides an in depth literature review of the research in the field of WSNs, RTI, and MTT. Chapter 3 describes the methodology used in the completion of this research. It also describes how all experiments are set up and how all data will be analyzed. Chapter 4 contains all experimental results in support of this research. Analytic results relative to the objective of the problem statement are presented in Chapter 4. Chapter 5 summarizes all the research conducted in this document and provides the conclusion of what work has been accomplished and the contributions of this research. Lastly, Chapter 5 describes additional research areas that can follow on to this research.

II. Related Work

This chapter provides an introduction and background to the theory behind RF-based localization methods. The research efforts and evolution of RF-based localization methods have provided the foundation for RTI. RTI is a DFL method that uses a WSN to geolocate the position of one or multiple targets. Geolocation involves using data from a sensor network to assess and estimate the location of a moving or stationary target. RSS, AoA, and/or TDoA measurements can be used to estimate target location in sensor networks [14], [15], [16]. RTI uses the RSS information from each radio to estimate the position of the target(s) [5].

2.1 Notational Conventions

Throughout the paper, $(\cdot)^{-1}$ and $(\cdot)^T$ denote a matrix inverse and transpose respectively. A hat (e.g. $\hat{x}$) indicates an estimate of its argument and a bar (e.g. $\bar{x}$) represents the ensemble or sample mean of the argument. All column vectors are indicated with **bold** lower case letters, row vectors are denoted with a transpose operator, and matrices are denoted by capital **BOLD** letters.

2.2 Radio Tomographic Imaging Background

RTI is an emerging concept that uses DFL and the RSS values of a WSN to geolocate a stationary or moving target. The WSN is set up around the AoI and the target of interest, which can be a person or object. The target inside the AoI creates a shadowing loss between each link being obstructed by the target [1].

2.2.1 Ultrawideband Imaging.

RTI is a derivative of RF-based radar applications from the commercial industry. From [1], UWB-based imaging devices have been developed by various companies which use phased array radars to estimate range and bearing. An UWB network consisting of multiple radar transmitter and receivers can be set up around a concentrated AoI to geolocate a target. This process can be referred to as *active localization* [17]. In order to estimate range and bearing, the devices emit UWB pulses to measure the echoes from the devices. Based on the estimates for the the change in range and bearing when a target is present, an image of the AoI can be estimated to show the estimated location of the target. The estimated image can be mapped to a pixel scene of the AoI to show the presence or absence of target(s) [18]. The benefits of UWB is that it is device free, can offer accurate position estimation of a target, and is passive [17], [18], [19]. The challenges with UWB is that it requires a large bandwidth, can be expensive, and suffers monostatic scattering losses over larger areas [5], [6].

2.2.2 Multiple-Input-Multiple-Output Radar.

Multiple Input, Multiple Output (MIMO) radar has been an emerging field that utilizes multiple radars transmitter and receivers to geolocate objects within an area in which the radars surround. MIMO is often referred as a type of multistatic radar. From [20], MIMO is used for target detection. The waveforms from the transmitters are scattered from the target and the receivers are able to resolve the waveforms to geolocate the target inside the spatial area. It has also been shown that MIMO can be used to track moving targets by computing the Doppler shift. RTI eliminates the need to measure reflections, but instead uses shadowing loss as the basis for the image reconstruction inside the AoI [21].

2.2.3 Device-Free Localization.

The access to and growing usage of Wireless Local Area Networks (WLAN) have allowed for the increase of DFL systems. Active based systems such as Global Positioning System (GPS), various RF based systems, Ultrasonic based systems, and Infrared (IR) based systems require a device attached to the target in some fashion in order to localize the target. DFL does not require an emitter from the target being tracked, thus is an unobtrusive way to estimate the position of a target. Observing changes in the RSS of a WLAN environment is a technique that can be used localize a target in a passive DFL environment [22], [23].

The growth of DFL systems and advancements in WLAN communication have provided the motivation to research DFL localization methods. Approximating a target's location has provided useful to applications such as, unobtrusively monitoring patients in a hospital, estimation location of assets, network access based on user's location, and indoor traffic monitoring [22].

2.2.4 Radio Tomographic Imaging with Received Signal Strength.

With the growth of low cost RFICs, RTI has been enabled to grow as an emerging technology in the realm of DFL. RTI uses RSS measurements from a RF network that is deployed around an area of interest. All the radios in the network are capable of receiving and transmitting with one another. The attenuation created by the objects or people inside the network are utilized to obtain images of the network area. Due to noise in the channels, noise models are investigated in the RTI system. Due to noise, regularization methods have been explored to estimate the image of the RTI network. Error bounds on the image can be used to calculate the accuracy of a particular RTI network [1].

Unique Links. Since all the radios in the RTI network can transmit and receive RSS among one another, the number of two-way unique links, M, can be calculated as,

$$M = \frac{N^2 - N}{2}, \tag{2.1}$$

where N is the number of radios in the RTI network [1]. Figure 2.1 is an illustration of all the links of a RTI network with $N = 36$ nodes and $M = 630$ links.

Received Signal Strength. RTI uses RSS to measure signal power from one radio to another in the network. From [5], the Received Signal Strength Indicator (RSSI) from the network is the only information needed to localize targets. The hardware can remain simple because no other information is needed in this RSS system. In the literature, RSS

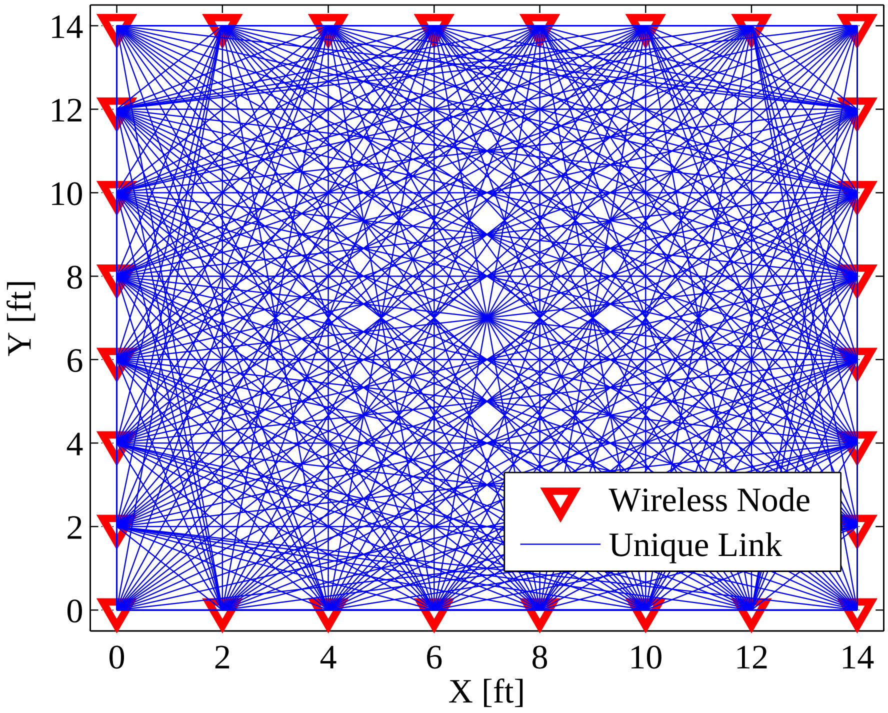


Figure 2.1: Illustration of the links created in a RTI network [1].

measurements are typically modeled as log-normal with a Gaussian distribution. The received power, P_l at each link, l over the wireless channel is [5], [24],

$$P_l \sim \mathcal{N}\left(\bar{P}(d_l), \sigma^2\right). \tag{2.2}$$

Path Loss Model. The RTI path loss model describes the RSS loss due to shadowing loss from objects, fading loss, static losses, and measurement noise for each link, l in the network. The RSS of any given link l, at time t, can mathematically be computed as [1], [10]:

$$r_l(t) = P_T - L_l(t) - S_l(t) - F_l(t) - v_l(t), \tag{2.3}$$

where

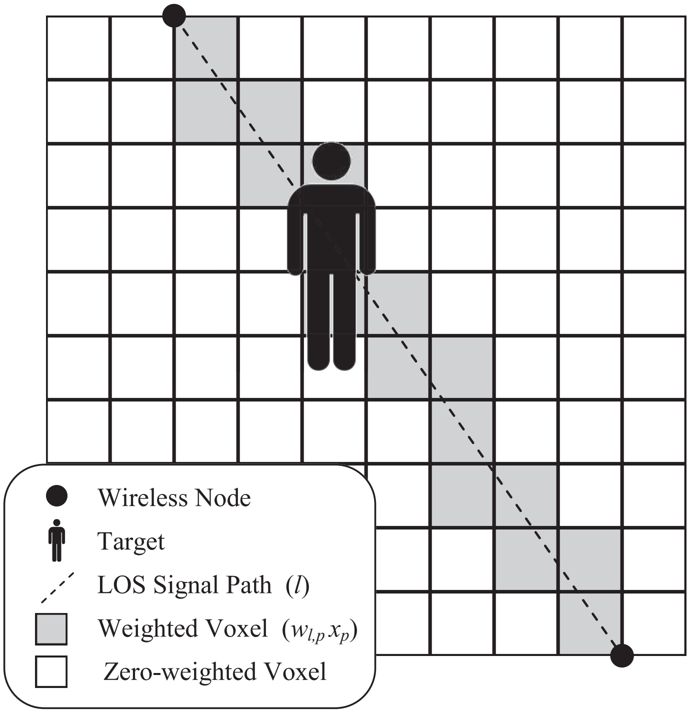


Figure 2.2: Illustration of a single obstructed link in a RTI network [1].

- P_T: Transmitted power (decibels (dB)).
- $L_l(t)$: Static losses due to distance, antenna patterns, device inconsistencies, etc (dB).
- $S_l(t)$: Shadowing loss due to objects attenuating the signal (dB).
- $F_l(t)$: Fading loss caused by constructive and destructive interference of narrow-band signals in multipath environments (Non-Shadowing Loss) (dB).
- $v_l(t)$: Measurement Noise (dB).

Radio Tomographic Imaging Linear Model. The entire vector of RSS links can be described in matrix form from the following linear model [1]

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{n}, \tag{2.4}$$

where $\mathbf{y}$ is the change in RSS from the baseline, which has length M. $\mathbf{W}$ is a weight matrix of dimension $M \times P$, where M corresponds to the number of links and P represents the number of pixels in the given RTI network. Each RSS measurements is measured in decibels (dB).

Noise. The noise, $\mathbf{n}$ from the model from (2.4) is typically modeled as Additive White Gaussian Noise (AWGN) [25], [26], [27], [28]. The noise can empirically be modeled as

$$\mathcal{N}\left(0, \sigma_l^2\right), \tag{2.5}$$

where σ_l^2 is the measured variance of the link data for the particular RTI network. As discussed in [1], the statistics of the noise vector must be examined. From [28], the main contributors to the noise, $\mathbf{n}$ is: the free space path loss, loss due to shadowing, receiver gains (which can be antenna gain and/or cabling losses), and transmitter gains. Hamilton also assumes a single-path propagation, but notes that it can be extended to multi-path channels. In [1], a Gaussian mixture model was used to fit the measured data. The two-part log-normal mixture model, with values in decibels, can be modeled as

$$f_{n_i}(u) = \sum_{j \in \{1,2\}} \frac{p_k}{\sqrt{2\pi\sigma_j^2}} \exp\left[\frac{u^2}{2\sigma_j^2}\right], \tag{2.6}$$

where $f_{n_i}(u)$ is the probability density function of the random noise variable n_i, p_j is the probability for part j, and σ_j^2 is the variance of part j. This model is based off the results from [29].

2.3 Weighting Models

If absolute knowledge of the area of interest was available, the weights for every link, l, at each pixel would be definitely known. In time critical situations where RTI would be utilized, users will likely not have the luxury of surveying the scene for all obstructions, interior arrangements, or have access to any other site specific information. This is why a statistical model for $\mathbf{W}$ needs to be robust enough in an array of different environments and network sizes. In the literature, various models have been explored to represent the weighting matrix $\mathbf{W}$ from (2.4). Although $\mathbf{W}$ has taken on various forms in the literature, the weighting matrix can be decomposed into two parts as shown in [30], [31]. The general form for $\mathbf{W}$ can be decomposed as

$$\mathbf{W} = \mathbf{S} \odot \mathbf{\Omega}, \tag{2.7}$$

where $\mathbf{S}$ is a binary selection matrix, $\odot$ is an element-wise Hadamard multiplication, and $\mathbf{\Omega}$ is a real-valued matrix of weights assigned to each pixel in the network. Using singular value decomposition (SVD), $\mathbf{W}$ can be also be represented as

$$\mathbf{W} = \mathbf{U\Sigma V^T} \tag{2.8}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, and Σ is a diagonal matrix of singular values [32]. The three weighting matrices commonly found in RTI literature are: the *NeSh Model* [1], the *Line Model* [31], [33], and the *NeSh-Line Model* [25], [26] with the *NeSh Model* being the most widely used. There are also other weighting models described in [23], [27], [28], [34], and [35], but they are not utilized as frequently in literature as the first three mentioned models.

2.3.1 NeSh Normalized Ellipse Model.

The *NeSh Model* was first used in [1], but has since been expanded in [36]. The expanded model has been used in [6], [37], [38], [39], [40]. The *NeSh Model* was designed to take into account the shadowing loss described in (2.3), on each link *l*. The adapted model uses a normalization factor to take into account that as the distance of links increase, the variance should not as well. The normalized factor from [36] is

$$\mathbf{\Omega}^{NeSh} = \frac{1}{\sqrt{d_l}}. \tag{2.9}$$

From [1], an ellipsoid with a focus at each radio location is used to determine the weighting for each link of the network. The *NeSh* weighting is described mathematically as

$$w_{l,p}^{NeSh} = \frac{1}{\sqrt{d_l}} \begin{cases} 1, & \text{if } d_1(l,p) + d_2(l,p) < d_l + \lambda, \\ 0, & \text{otherwise,} \end{cases} \tag{2.10}$$

where d_l is the distance or length of link l, $d_1(l, p)$ is the distance from the first node of link l to the location of pixel p and $d_2(l, p)$ is the distance from the second node of link l to to the location of pixel p. λ is a tunable parameter which represents the width of the ellipse. If the sum of d_1 and d_2 is less than the length of the link d plus the tunable parameter λ, a 1 is assigned in the binary selection matrix, else, a 0 is assigned. Using the general decomposed form for **W** from (2.7), the decomposed model for **W** is

$$\mathbf{W}^{NeSh} = \mathbf{S}^{Ellipse} \odot \mathbf{\Omega}^{NeSh}. \tag{2.11}$$

This model puts the weight only on pixels that fall within the ellipsoid computed for each link. However, this model assumes that all pixels that fall within the ellipsoid have equal weight.

2.3.2 Line Model.

The *Line Model* originated from [33], [41], but have been utilized in RTI applications in [26], [31], [42]. $\mathbf{\Omega}^{Line}$ is only concerned with the portion of link l that passes through pixel p. The binary selection matrix, $\mathbf{S}^{Line} = 1$ if link l traverses through pixel p, else $\mathbf{S}^{Line} = 0$. Therefore, the weighting matrix for the *Line Model* can mathematically can be computed as

$$w^{Line}_{l,p} = L_{l,p} \begin{cases} 1, & \text{if link } l \text{ pixel } p, \\ 0, & \text{otherwise}, \end{cases} \tag{2.12}$$

where $L_{l,p}$ is the length of the portion of link l that traverses through pixel p. The weight of each entry in the matrix is assigned based on the length of the link through the pixel rather than the square root of the distance of the entire link as shown in the *NeSh Normalized Ellipse Model*. Similar to (2.7), **W** can be decomposed as

$$\mathbf{W}^{Line} = \mathbf{S}^{Line} \odot \mathbf{\Omega}^{Line}, \tag{2.13}$$

where the *Line Selection Matrix* is described more in depth in [33], [42], [41]. The *Line Model* is simple to implement and in [31], the *Line Model* is described as being the more computationally efficient model over the other commonly used weighting models used in RTI.

2.3.3 NeSh-Line Model.

The *NeSh-Line Model* is a hybrid of the *NeSh* and *Line* models previously discussed in this section. This model was first used in [25], [26]. The *Line Selection Matrix*, $\mathbf{S}^{Line}$ is the same one found in the decomposed *Line Model*. The weighting factor, $\mathbf{\Omega}$ is computed by calculating the link distance of l, similar to the *Line Model*, and the inverse of the square root of the distance of the link that traverses through pixel p, which is similar to the *NeSh Model*. $\mathbf{W}$ for the *Nesh-Line Model* can be decomposed as

$$\mathbf{W}^{NeShLine} = \mathbf{\Omega}^{NeShLine} \odot \mathbf{S}^{Line}, \tag{2.14}$$

where

$$w_{l,p}^{NeShLine} = \frac{L_{l,p}}{\sqrt{d_l}} \begin{cases} 1, & \text{if link } l \text{ traverses pixel } p. \\ 0, & \text{otherwise.} \end{cases} \tag{2.15}$$

2.4 Regularization Methods

Since the output from the RTI network, $\mathbf{y}$ from (2.4) is the only output given from the network, the image scene, $\mathbf{x}$ from (2.4) needs to be estimated. Wilson and Patwari discuss different methods to estimate $\mathbf{x}$ in [1], [32]. Since the goal is to estimate $\mathbf{x}$ and minimize the noise in the least-squared error sense, various regularization methods have been explored.

Ill-Posed Inverse Problem. The linear model from (2.4) is common in other physical problems [1], where the goal would be to minimize the noise in the least-squared sense which can mathematically be represented as

$$\hat{\mathbf{x}}_{LS} = \arg\min_{x} ||\mathbf{W}\mathbf{x} - \mathbf{y}||^2. \tag{2.16}$$

Using to (2.16), the least-squared solution for (2.4) would be

$$\hat{\mathbf{x}}_{LS} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{y}. \tag{2.17}$$

However, $\mathbf{W}$ in most cases will not be full rank, thus estimating $\mathbf{x}$ is an ill-posed inverse problem. Due to the transfer matrix, $\mathbf{W}$ having much smaller values than the measurement noise, regularization is useful in estimating $\mathbf{x}$. Without any type of regularization, the measurement noise would be amplified when solving for $\mathbf{x}$ [32]. Below are a few popular regularization methods researched for RTI.

2.4.1 Tikhonov Regularization.

Tikhonov regularization is the most widely used in RTI. It is a popular regularization because it forces a solution by adding an energy term [1], [32]. This regularization has the flexibility to manipulate the desired output by picking the regularization parameter α. The solution is outputed after the linear transformation of the measurement data [32]. The resulting objective cost function is

$$f_{TIK}(x) = \frac{1}{2}||\mathbf{W}\mathbf{x} - \mathbf{y}||^2 + \alpha\left(||\mathbf{D}_{\mathrm{x}}\mathbf{x}||^2 + ||\mathbf{D}_{\mathrm{y}}\mathbf{x}||^2\right), \tag{2.18}$$

where $\mathbf{D}_x$ and $\mathbf{D}_y$ are difference operators in the x and y directions of $\mathbf{x}$ respectively. To find the estimated scene $\hat{\mathbf{x}}$, the derivative of (2.18) needs to be set equal to zero which is

$$\hat{\mathbf{x}}_{TIK} = \underset{x}{\operatorname{argmin}}\left(\frac{1}{2}||\mathbf{W}\mathbf{x} - \mathbf{y}||^2 + \alpha\left(||\mathbf{D}_{\mathrm{x}}\mathbf{x}||^2 + ||\mathbf{D}_{\mathrm{y}}\mathbf{x}||^2\right)\right), \tag{2.19}$$

$$\hat{\mathbf{x}}_{TIK} = \left(\mathbf{W}^T\mathbf{W} + \alpha\left(\mathbf{D}_{\mathrm{x}}^T\mathbf{D}_{\mathrm{x}} + \mathbf{D}_{\mathrm{y}}^T\mathbf{D}_{\mathrm{y}}\right)\right)^{-1}\mathbf{W}^T\mathbf{y}. \tag{2.20}$$

In [1], the derivative operators are summarized by the *Tikhonov* matrix $\mathbf{Q}$. Substituting $\mathbf{Q}$ into (2.20) yields

$$\mathbf{Q} \triangleq \mathbf{D}_{\mathrm{x}}^T\mathbf{D}_{\mathrm{x}} + \mathbf{D}_{\mathrm{y}}^T\mathbf{D}_{\mathrm{y}}, \tag{2.21}$$

$$\hat{\mathbf{x}}_{TIK} = \left(\mathbf{W}^T\mathbf{W} + \alpha\mathbf{Q}\right)^{-1}\mathbf{W}^T\mathbf{y}. \tag{2.22}$$

In matrix form, the linear operator on $\mathbf{y}$ can be demonstrated by

$$\hat{\mathbf{x}}_{Tik}{=}\mathbf{\Pi}_{Tik}\mathbf{y}, \tag{2.23}$$

where

$$\mathbf{\Pi}_{Tik} = \left(\mathbf{W}^T\mathbf{W} + \alpha\left(\mathbf{D}_{\mathrm{x}}^T\mathbf{D}_{\mathrm{x}} + \mathbf{D}_{\mathrm{y}}^T\mathbf{D}_{\mathrm{y}}\right)\right)^{-1}\mathbf{W}^T. \tag{2.24}$$

2.4.2 Truncated Singular Value Decomposition.

Another popular regularization method is Truncated Singular Value Decomposition (TSVD) [32]. The objective is to remove smaller singular values from the weighting matrix $\mathbf{W}$. This method is similar to scalar regularization [43], where only $g < N$ singular values from Σ are used in the reconstruction. The linear transformation matrix is given by

$$\mathbf{\Pi}_{TSVD} = \sum_{i=1}^{g<N} \frac{1}{\sigma_i} \mathbf{u}_i \mathbf{v}_i^T = \mathbf{U}_k \Sigma \mathbf{V}_k{}^T, \tag{2.25}$$

where $\mathbf{U}$, $\mathbf{V}$, and Σ are matrices described in section 2.3. Therefore, the image estimate using TSVD regularization is

$$\hat{\mathbf{x}}_{TSVD} = \mathbf{\Pi}_{TVSD} \mathbf{y}. \tag{2.26}$$

The drawback to this method is that since the singular vectors are dependent on the node locations, TSVD lacks the ability to incorporates the parameter α to force desired properties of the image estimate. However, like *Tikhonov* regularization, the transformation matrix, $\mathbf{\Pi}_{TSVD}$ can be pre-calculated prior to recording data for quick and real-time applications. The results from [32] show that *Tikhonov* and *Total Variation (TV)* do a better job in minimizing noise present in the image estimate. This is due to the high frequency components that are included in the reconstruction.

2.5 Node Density

The node density of a RTI network can greatly affect the accuracy of the image scene $\mathbf{x}$. The more dense a network is, the more likely the accuracy would be higher than a network with a sparse amount of motes further apart. The more links that pass through a particular area, the more RSS values would be present to estimate the image scene [1].

Cramer-Rao Lower Bound. Wilson and Patwari derive the Cramer-Rao Lower Bound (CRLB) for the unbiased estimator $\hat{\mathbf{x}}_{Tik}$ [1]. The CRLB is the error bound at each pixel location p of the particular RTI network. The Mean Squared Error (MSE) bound for a RTI network is given by

$$COV(\mathbf{x}_{Tik}) \geq \left(\gamma \mathbf{W}^T \mathbf{W} + \mathbf{C}_x^{-1}\right) \tag{2.27}$$

where γ is computed by the following integration

$$\gamma = \int_{-\infty}^{\infty} \frac{f_n'(u)^2}{f_n(u)} du. \tag{2.28}$$

f_n is the two-component Gaussian distribution of n found in (2.6). $\mathbf{C}_x$ is the spatial covariance model used in [44]. The spatial coveriance model is computed by

$$[\mathbf{C}_x]_{p,q} = \sigma_x^2 e^{\frac{-d_{p,q}}{\delta_c}}, \tag{2.29}$$

where ${\sigma_x}^2$ is the variance at each pixel, $d_{p,q}$ is the distance from each pixel p to pixel q, and δ_c is the correlation parameter.

Cylindrical Human Model. In [1], Wilson and Patwari use a Cylindrical Human Model to assess the accuracy of a given RTI network. The purpose is to assess the "true" attenuation field to the image scene being estimated. The model assumes a uniform attenuation throughout the radius R_h, of a human positioned at a coordinate location $\mathbf{C}_h$. The model for the Cylindrical Human Model image scene $\mathbf{x}_h$ can be described as

$$\mathbf{x}_h = \begin{cases} 1, & \text{if } \left\|\mathbf{x}_p - \mathbf{C}_h\right\| < R_h, \\ 0, & \text{otherwise,} \end{cases} \tag{2.30}$$

where $\mathbf{x}_p$ is the (x, y) center of pixel p.

Spherical Model. In [42], Martin *et al.* propose a spherical model to represent a spherical obstruction. The obstruction in $\mathbf{x}$ is

$$\mathbf{x}_p = A \exp\left(\frac{-1}{2r_o^2} ||x(p) - c_o||^2\right), \tag{2.31}$$

where A is attenuation (dB) per voxel of obstruction, r_o is the defined radius of the obstruction, γ is derived from the noise model found in (2.28), and c_o is the coordinate position of the obstruction. The link passing through the center of the obstruction would have a link attenuation of

$$y_l = \sum_p x_p \approx \int \frac{x_p dx}{\delta} = \frac{A\sqrt{2\pi}r_o}{\delta}. \tag{2.32}$$

2.6 Radio Tomographic Imaging Methods

Since the creation of RTI, other methods of RTI have been explored to improve tracking and estimation of targets in RSS-based, DFL networks. The various methods were created to improve accuracy in an array of different applications. For example, different methods of RTI can be utilized to track moving targets versus stationary targets. The WSN may be Line-of-Sight (LOS) or may need to be set up with obstructions between the sensors and the target(s). The different RTI methods have varying capabilities that can be applied to the applicable target tracking situation.

2.6.1 Mean-Based Radio Tomographic Imaging.

Mean-based Radio Tomographic Imaging (MRTI) is a commonly used method in the literature and is one of the simplest to implement in terms of complexity. This method is also referred to as shadowing-based RTI as it quantifies the loss on each link affected by the target to localize the target's location [1]. Mean-based or shadowing-based RTI is typically referred to as RTI [5], [31], [38], [2], [45].

Measurement Model. The shadowing loss R on each link l can be approximated by a sum of the attenuation that occurs at each pixel in the network. For N frames, the link RSS at frame n can be mathematically described as

$$\bar{r}_{l,n} = \frac{1}{N}\sum_{i=0}^{N-1} r_l\left(t_n - i\right), \tag{2.33}$$

$$\Delta\bar{r}_{l,n} = \bar{r}_{l,n} - \bar{r}_{l,c}, \tag{2.34}$$

where $\bar{r}_{l,c}$ is the calibration RSS on each link. Therefore the sample mean for each link in vector notation is

$$\mathbf{y}_{mean} = \left[\Delta\bar{r}_{1,n}, \Delta\bar{r}_{1,n}, ..., \Delta\bar{r}_{K,n}\right]^T. \tag{2.35}$$

As discussed in section 2.4, the inverse problem given **y** needs to be solved to estimate **x**. Using the sample mean from (2.35) is useful in locating both static and moving targets. Due to the noise caused by walls or similar solid foundations, this method is better suited for LOS applications rather than non-LOS applications [36], [37], [44]. Additionally, since mean-based RTI only uses the change in the mean RSS, quick or sporadic movement would degrade the accuracy of the network [2].

2.6.2 Variance Radio Tomographic Imaging.

Variance-based Radio Tomographic Imaging (VRTI) utilizes the variance between RSS frames from the RTI network to estimate the target's location. Due to only the variance being needed to estimate the image from the attenuation field, the need to calibrate the RTI network prior to taking data measurements is alleviated [6], [38], [39].

The VRTI system uses a vector **y** of RSS measurements on M links in the RTI network to determine the variance between each frame, where the variance is measured in dB. The RSS variance of on each link can mathematically be defined as

$$VAR[\mathbf{R}_{dB}] = \sum_{p} w_{p,l}x_p + n_l, \tag{2.36}$$

where n is the measurement noise and modeling error, w_p is the variance caused by a movement in pixel p, and R_{dB} is the received signal strength. Using the linear model from (2.4), the linear system for VRTI can be expressed as

$$\mathbf{s} = \mathbf{W}\mathbf{x} + \mathbf{n}, \tag{2.37}$$

where **s** is an $M\times 1$ measurement vector of the variance of each link l, **W** is a chosen weight matrix as discussion in section 2.3, and **x** is the $N \times 1$ scene image that is estimated using a chosen method as discussed in section 2.4. In [38], Tikhonov regularization is used for optimization of the image estimate and is defined as

$$\hat{\mathbf{x}}_{TIK} = \mathbf{\Pi}_{TIK}\mathbf{s}, \tag{2.38}$$

where $\mathbf{\Pi}_{Tik}$ is from (2.24).

Results. VRTI improves imaging in through-wall applications as shown in [6], [38], [39]. In [38], imaging through the wall with mean-based RTI was compared to VRTI to show that VRTI had a less noisy image through the walls. VRTI is valuable because locating moving targets outside the walls is extremely valubale for police, military, and rescue teams to get an image of targets inside a building prior to entering [10]. Kalman filters can be utilized to further increase the accuracy of target tracking of moving targets using VRTI. Since VRTI utilizes the changes in variance on the links in the network for imaging rather than the change in mean of the static RSS losses, VRTI is not as viable for locating stationary targets as MRTI.

2.6.3 Kernel Radio Tomographic Imaging.

Kernel-based Radio Tomographic Imaging (KRTI) compares the short-term and long-term histograms of a RTI network to locate the position of any targets inside the network. This method has the benefit of locating both stationary and moving targets in LOS and non-LOS environments. With this method, a training period is required to record the RSS histograms on all the links in the network. Unlike MRTI or VRTI, the objective is to quantify the change in RSS in the network caused by a person through the use of histograms rather than the mean or variance of the link RSS [36], [38], [39]. Zhao *et al.* find the distance between short-term and long-term histograms in a RTI network using the Kullback-Leibler divergence [2].

Distance Between Long-Term and Short-Term Histograms. In KRTI, every link l is characterized by short-term and long-term histograms of past RSS measurements. At frame n, the weighted average of the histogram h is

$$\mathbf{h}^n = \sum_i w_{n,i}\mathbf{I}_{y^i}, \tag{2.39}$$

where **I** is an N-length indicator vector and y^i is the RSS at time i. The exponentially weighted moving average is given by

$$w_{n,i} = \begin{cases} \beta(1-\beta)^{n-i}, & i \leq n, \\ 0, & \text{Otherwise}, \end{cases} \tag{2.40}$$

where β is the forgetting factor, $0 < \beta < 1$. A higher β increases the importance of the most recent RSS measurements, while a lower β would be more appropriate for long-term histograms. The kernel distance is between the long-term and short-term histograms is found by

$$D_K(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T\mathbf{K}\mathbf{p} + \mathbf{q}^T\mathbf{K}\mathbf{q} - 2\mathbf{p}^T\mathbf{K}\mathbf{q}, \tag{2.41}$$

where **K** is the kernel, **p** is the short-term histogram and **q** is the long-term histogram defined in [2]. A commonly used kernel is the Epanechnikov kernel, which minimizes the integrated squared error [2] and is defined as

$$\mathbf{K}(y_i, y_j) = \frac{3}{4}\left(1 - \frac{\left|y_i - y_j\right|^2}{\sigma_E^2}\right)\mathbf{I}_{|y_i - y_j| \leq \sigma_E^2}, \tag{2.42}$$

where i and j are elements of the RSS links in y, **I** is the indicator function and σ_E^2 is the Epanechnikov kernel width.

Kernel Distance-Based Radio Tomographic Image Formation. Once the histogram distances are computed, $\mathbf{d} = [\mathbf{d}_1, ..., \mathbf{d}_M]^2$ can denote the histogram difference of all links, M histogram differences, where $\mathbf{d}_1 = D(\mathbf{p}_1, \mathbf{q}_1)$. Using the RTI linear model from (2.4), **d** is defined as

$$\mathbf{d} = \mathbf{W}\mathbf{x} + \mathbf{n}, \tag{2.43}$$

where **n** is the noise vector and **W** is a chosen weight model discussed in Section 2.3 [1], [32], [38], [42], [46]. The histogram difference vector, **d** is used to form the image $\hat{\mathbf{x}}$, which has the modified least-squares solution

$$\hat{\mathbf{x}} = \left(\mathbf{W}^T\mathbf{C}_n^{-1}\mathbf{W}^T + \mathbf{C}_x^{-1}\right)^{-1}\mathbf{W}^T\mathbf{C}_n\mathbf{d}, \tag{2.44}$$

where $\mathbf{C}_x$ is the covariance matrix of $\mathbf{x}$ and $\mathbf{C}_n$ is the covariance matrix of each link's measurement noise [2]. Using Tikhonov regularization [21], the matrix notation of the modified least-squares formulation is

$$\hat{\mathbf{x}}_K = \mathbf{\Pi}_K \mathbf{d}, \tag{2.45}$$

where

$$\mathbf{\Pi}_K = \left(\mathbf{W}^T\mathbf{W} + \sigma_n^2 \mathbf{C}_x^{-1}\right)^{-1} \mathbf{W}^T, \tag{2.46}$$

and the variance of the measurement noise is given by σ_n^2.

Results. In [2], KRTI is used to track a moving target with the addition of a Kalman filter. The transition model of the Kalman filter includes the target's location and velocity. A variety of experiments were performed which included a bookstore environment which had bookshelves as obstructions and a large living room in a residential setting. The experiments used 34 radios with twenty locations being estimated in each experiment. The overall average error at each estimated location $\hat{z}_i$ was calculated by

$$\bar{e} = \frac{1}{20} \sum_{i=1}^{20} ||\hat{z}_i - z_i|| \tag{2.47}$$

where z_i is the true location. In the experiments performed, it was found that KRTI had a lower average location error than VRTI and Sub-VRTI [6], [37], [38]. Overall, KRTI offered over a 30% improvement over VRTI and Sub-VRTI [2]. Stationary experiments were also performed and had an average location error of less than 0.81 meters.

2.6.4 Other Radio Tomographic Imaging Methods.

There have been other methods of RTI that have been explored in literature. The other methods described in this section utilize a channel that is chosen by the user. In [40], Kaltiokallio *et al.* propose a method to select an optimal channel so that the reliability of the links is maximized. Experimental results show that channel diversity can increase the accuracy of a network. Histogram-based Radio Tomographic Imaging (HRTI) was first demonstrated in [47] and is the foundation for KRTI which uses the distances between

Table 2.1: Radio Tomographic Imaging features [2].

Features	**RTI**	**VRTI**	**KRTI**
Through wall?	Yes	Yes	Yes
Calibration?	Yes	N/A	No
Stationary Targets?	Yes	No	Yes
Real-Time?	Yes	Yes	Yes

link histograms to estimate the image. In [48], the use of directional antennas is used in the sensors of a network. This method is known as Direction-based Radio Tomographic Imaging (DRTI) and was proposed to improve the localization accuracy of RTI. Since the number of pairs grow quite large with a greater number of sensors, a lower number of sensors in the network would need to be used for this to be feasible. However, the experimental results showed that DRTI can improve the accuracy over omni-directional antennas in both LOS and non-LOS environments.

2.6.5 Radio Tomographic Imaging Features.

Since there are multiple RTI methods that have been explored, each one has features available that could be appropriate for different situations. Table 2.1 shows various features for shadow-based RTI, VRTI, and KRTI. In settings where imaging will need to be done through the wall, VRTI or KRTI would be the preferred methods over shadow-based RTI [2]. Additionally, VRTI and KRTI do no require calibration, but RTI does. This could be a drawback in an emergent situation where taking the time to calibrate may not be feasible. MRTI has the benefit of estimating the location of stationary targets over VRTI. When the target is stationary, there would not be a significant variance in the links of the network to accurately locate the target [6], [39].

2.7 Localization Methods

Due to RTI being an ill-posed inverse problem, there is not a stable and unique solution to the least-squares formulation for $\mathbf{x}$. Once a regularization and estimator is chosen, $\hat{\mathbf{x}}$ needs to be analyzed to locate the target(s) of interest. Depending on how many targets are in the scene, there are two commonly used estimators in RTI applications to locate the target(s) [12], [38], [2], [49].

Maximum A Posterior Estimation. For a single target application, a Bayesian statistic on the estimated scene, $\mathbf{x}$ can be applied to estimate the location of the target [49]. The MAP estimated location would be the pixel with the maximum value. The mathematical notation for this estimate is

$$\hat{z} = \arg\max_{p} \hat{x}_p, \tag{2.48}$$

where $\hat{z}$ is the estimated location of the target and $\hat{x}_p$ is the pixel intensity at each pixel p.

K-Means Clustering. K-means clustering is a popular data mining tool to find patterns or clusters of interest from a set of data. It is popular among machine learning applications, pattern recognition, hyper-spectral imagery, artificial intelligence, crowd analysis, and MTT [11], [12], [13]. The K-means algorithm clusters a given set of data together into K partitions with the goal of minimizing the variance of each cluster. This algorithm is similar to the expectation-maximization algorithm where the end goal is to find the optimal center of the defined number of clusters [13]. K-means is an iterative process where the objective is to minimize the total inter-cluster variance. This process assumes a fixed a priori, K for the number of clusters to be found from the given data set. Therefore, the objective function, J is a squared error function which is

$$J = \sum_{i=1}^{K} \sum_{x_j \in S_i} \left\| x_j - C_i \right\|^2, \tag{2.49}$$

where $\mathbf{x}_j$ is the set of data to be separated into clusters, K is the number of clusters, n is the number of cases, S_i is the set of pixels assigned to cluster i and C_i is the centroid for cluster i [13]. The algorithm is performed by the following steps [50]:

1. Place K points into the spatial area represented by the points that are being clustered; the points represent the initial clusters and centroids.

2. Assign each object to the group with the closest centroid using the squared error function from (2.49).

3. When all objects have been assigned, recalculate the positions of the K centroids.

4. Repeat steps 2 and 3 until the centroid positions converge.

Separation of all points in the data set is obtained when all objects are assigned to to a cluster by minimizing the euclidean distance of all the points in the data set to the cluster centroid. This process can be extremely fast, since in practice, it is repeated less than n times [13].

There are drawbacks to this iterative process. In terms of performance, K-means does not guarantee to return a global optimum. Since the heuristic algorithm described in this section starts with a random initialization, the final solution is sensitive to the initial set of clusters. If the number of K values is inappropriately chosen, the algorithm can produce poor results. Therefore, the algorithm relies heavily on picking a value of K that would yield desired results [13]. In [50], Chen and Shixiong propose an improved method to pick the initial centers of the clusters. This method proposes picking initial centroids already close to large quantities of points.

2.7.1 Multiple Target Tracking.

Although single targets are mainly used in literature [6], [31], [42], [51], MTT in RTI has started to be explored in [9], [10], [11]. Bocca *et al.* explore real time tracking with multiple targets using RTI [8]. Channel diversity from [40] is utilized in conjunction with machine vision methods to track multiple targets. The results presented are from an open environment, a one bedroom apartment, and a crowded office environment to demonstrate the capability to perform MTT with obstructions.

Pixel Threshold. When multiple targets are present, there are blobs of pixels in the image scene that go through a clustering process to estimate the location of the targets. Dynamic thresholding is used to reduce the size of the pixels that go through the clustering process. In [8], an algorithm is used to threshold pixels prior to being clustered. In an empty network, the average maximum intensity of the formed RTI images is used as the baseline, $\bar{I}_e$. The threshold is set to $2\bar{I}_e$ in order to disregard the pixels with low intensity. When targets are being tracked, the minimum intensity, I_{min} for targets $T = (t_1, ..., t_{|T|})$ is defined as

$$I_{\min} = \min_{t \in T} [\hat{\mathbf{x}}_G]_t, \tag{2.50}$$

where $\hat{\mathbf{x}}_G$ is the image scene after being filtered through a low-pass Gaussian kernel $\mathbf{G}$. The filtered RTI image $\hat{\mathbf{x}}_G$ is calculated as

$$\hat{\mathbf{x}}_G = \hat{\mathbf{x}} * \mathbf{G}, \tag{2.51}$$

where $\mathbf{G}$ is the Gaussian kernel and $*$ is the convolution operator. The Gaussian kernel is defined as

$$G(x, y) = \frac{1}{2\pi\sigma_G^2} \exp\left[-\frac{x^2 + y^2}{2\sigma_G^2}\right], \tag{2.52}$$

where σ_G is the standard deviation of the Gaussian kernel which is set to be 1 meter [8].

2.8 Chapter Summary

This chapter explained the various of forms of geolocation, which have led to the work accomplished in DFL. The background of RTI was discussed in this section as well as the the various RTI methods. In addition, the signal processing models, noise models, regularization, and weighting models have provided the foundation for the different forms of RTI found in the literature. Once an image is estimated from the information received from the network, a method such as a MAP estimator or K-means needs to be applied to estimate the location of the target(s). The weighting model, measurement model, and regularization used in this research have been described in this chapter. Since MTT is the focus of this research, this chapter described the K-means algorithm and how it can be applied to MTT.

III. Methodology

This chapter describes the methodology used in this research to establish and collect data from a RTI network comprised of multiple RFIC motes. The following sections outline the hardware and tools used in all the data collections. The system model and implementation of the research will be described in this chapter. Truth data using simulation and experimentation will be used to get a baseline performance of the network. Lastly, the methods in which the data will be analyzed post-data collection will be outlined in this section.

For all data collections, feet (ft) will be used as the metric for distance. All RSS values will be assumed to be in dBm. For the two-dimensional (2-D) RTI network, the pixel size will be $\Delta_p \times \Delta_p$. The $x - y$ plane will be utilized to show the 2-D pixel layout of all images. Therefore, all position and tracking coordinate estimation will be denoted by an (x, y) coordinate in feet.

3.1 Equipment and Tools

The equipment used in this research includes the Memsic TelosB mote platform [3] and a computer with *Microsoft Windows*® *7* for data collection and processing. The tools that were used in this research are described and listed below. Data collection, simulation, and analysis of all data were completed in `MATLAB`®.

Memsic TelosB Mote Platform. The wireless radios used in this research are made by Crossbow Technology Incorporated (Inc.) based in San Jose, California. The model utilized in the research is the TelesB mote TPR2400. University of California (UC) Berkeley developed the the open-source radios and is compatible with TinyOS distrubution. TPR2400 was developed for the research community and provides the users with the capability to interface with additional devices. The radios offer programming and data collection through a Universal Serial Bus (USB) interface.

Table 3.1: Select TPR2400 specifications [3].

Module	
RAM	10K bytes
Current Draw	1.8 mA
RF Transeiver	
Frequency Band	2400 Mhz to 2483.5 Mhz
RF Power	-24 dBm to 0 dBm
Outdoor Range	70 m to 100 m
Curend Draw (Receive Mode)	23 mA
Electromechanical	
Size	2.55 x 1.24 x .24 inches
Weight	0.8 ounces
User Interface	USB

Cygwin. The motes were programmed using *Cygwin* on a *Microsoft Windows® 7* 64-bit machine. *Cygwin* is a collection of GNU and open-source tools. It is a Unix-like environment which is used to interface with *Microsoft Windows® 7.* Cygwin was originally developed by Cygnus Solutions, but has been acquired by Red Hat [52].

TinyOS. *Tiny Operating System (OS)* is an open-source operating system designed for low-power wireless devices. The TelosB motes were equipped with *Tiny OS* which is written in NesC [53]. *TinyOS* includes the program file titled “BaseStation,” for programming the mote acting as the network *BaseStation*. Any mote can act as either a wireless radio in the network or the base station, but this is specified when programmed.

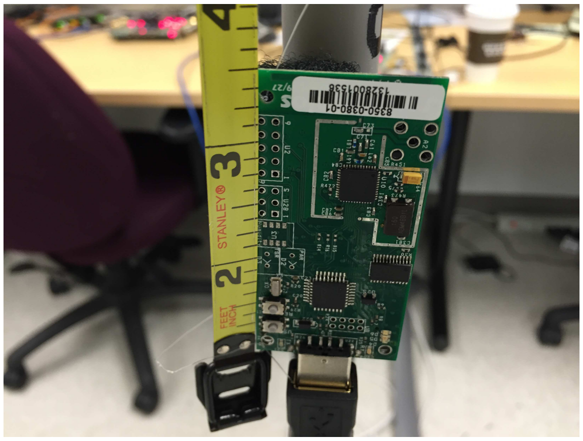


Figure 3.1: TelosB Mote.

Spin. The motes surrounding the AoI as well as the base station were programmed with the “Spin” protocol, created by the Sensing and Processing Across Networks (SPAN) lab at the Department of Electrical and Computer Engineering at the University of Utah. *Spin* is an open-source *TinyOS* program written in NesC. This program has the function of collecting RSS information from a WSN using a token passing protocol. *Spin* has specifically been tested with TelosB nodes. With this token passing protocol, only one radio transmits at a time through the channel; the motes transmit in the order specified by the user. For more information or to download the *Spin* program, refer to [54].

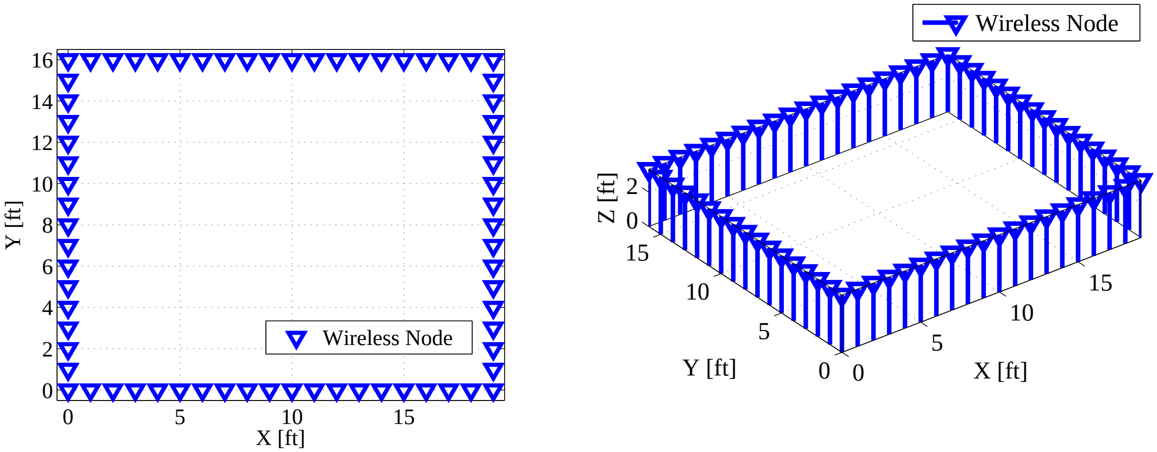
RTI LINK GUI. Data was collected using the *RTI LINK Graphical User Interface (GUI)* with the initial version created by Mr. Alex Folkerts (Southwestern Ohio Council for Higher Education (SOCHE) Intern), Mr. Tyler Heinl (SOCHE Intern), and Dr. Richard K. Martin (Associate Professor of Electrical Engineering at the Air Force Institute of Technology (AFIT)). The *RTI LINK GUI* is a MATLAB® based application designed to collect and save package data from the RTI network. The GUI receives the raw RSS data through the base station two’s complement and converts the values to hexadecimal. The collection of each link’s RSS values at each frame n is considered the vector $\mathbf{y} = [y_1, y_2, \cdots, y_M]^T$ from section 2.2.4. For MRTI, the GUI is capable of taking the raw $\mathbf{y}$ data at each frame and subtracting the calibration data to provide the mean change in RSS. The line weighting matrix, $\mathbf{W}$ from section 2.3 and the Tikhonov Pi matrix, $\mathbf{\Pi}_{Tik}$ from section 2.4 can be calculated prior to recording data to save computational time. User specified parameters such as the tunable parameter α, pixel size, Δ_p are inputted prior to calculating the weighting matrix and $\mathbf{\Pi}_{Tik}$ matrix. The GUI collects the raw link data at each frame in real time and uses the $\mathbf{\Pi}_{Tik}$ matrix as a linear operator to output the estimated image $\mathbf{x}$ in near real time. The calibration data and final recorded data can be saved in the form of raw link RSS data to provide the flexibility to compare different user parameters such as pixel size and regularization values.

3.2 Network Setup

All experimental data in this research was taken from the same RTI network. The network covered a 19 ft × 16 ft area surrounded by $N = 70$ motes described in Section 3.1. All the motes were placed 1 ft apart around the parameter of the network area. The motes were mounted on stands made from Polyvinyl Chloride (PVC) all at a height of 3.33 ft. The height of the sensors was chosen to be near the midsection of most adults. Inside the network, painters tape was used to mark off coordinates so that the true position of targets inside the network could easily be known. An illustration of the mote topology is shown in Figure 3.2. The number of unique links can be determined by (2.1). Therefore, the number of unique links for $N = 70$ nodes is

$$M = \frac{N^2 - N}{2} = \frac{70^2 - 70}{2} = 2,415 \text{ links.} \tag{3.1}$$

Figure 3.3 illustrates $M = 2415$ links for the experimental RTI network. The sensors were equally spaced apart to maximize the accuracy consistency throughout the entire network. In addition, the more nodes used, the more RSS link information would be available to estimate the image $\mathbf{x}$. Figures 3.4a and 3.4b are pictures of the network from two different corners.



(a) Aerial view of mote locations

(b) 3-D View from (0,0)

Figure 3.2: Aerial and three-dimensional views of mote topology.

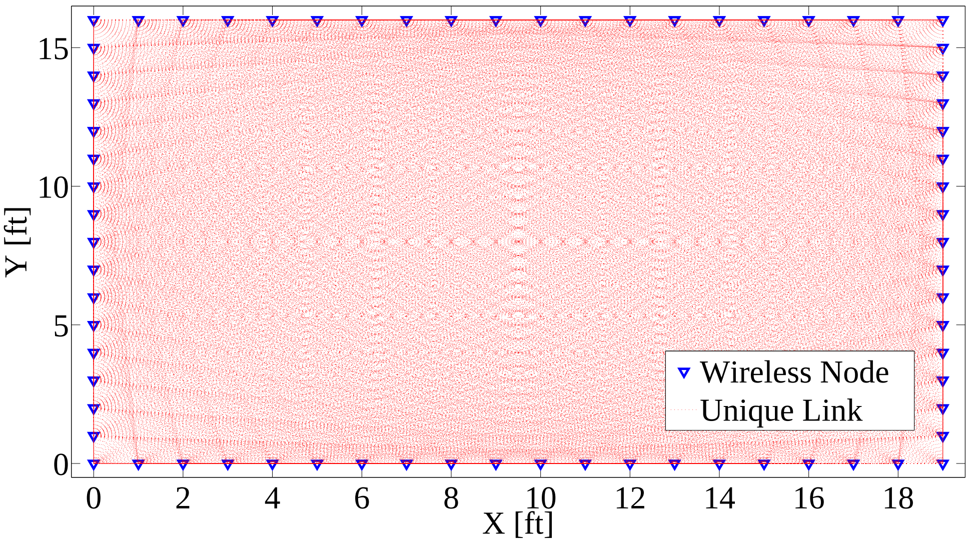
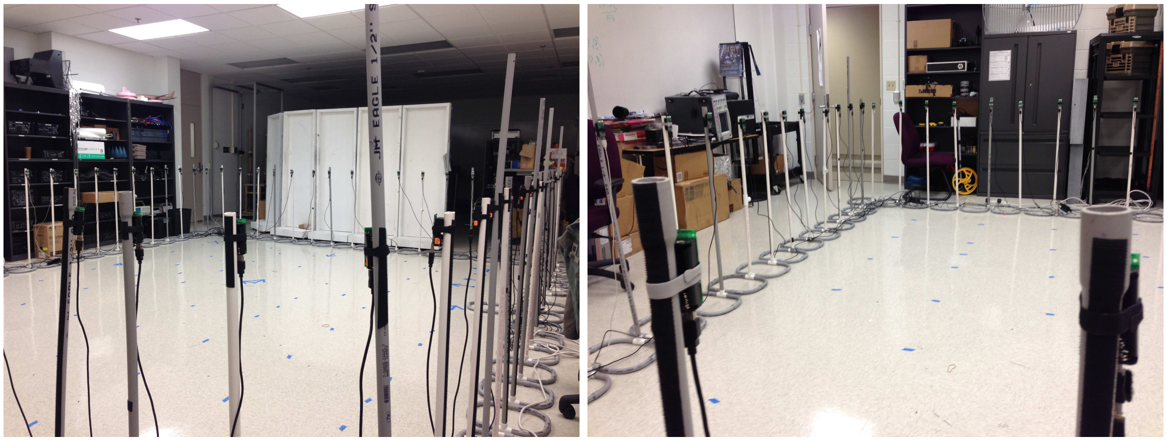


Figure 3.3: Experimental setup with $M = 2415$ links.

Radio Orientation. The TelosB TPR2420 are equipped with omni-directional antennas [55]. However, to be consistent, all motes were oriented in the same manner. The motes were positioned vertically with the USB interface facing towards the ground.



(a) View from (19,16)

(b) View of (19,0)

Figure 3.4: Pictures of the RTI experimental network.

Human Subjects. Human subjects were used in this research. Required human subjects training has been completed by the principal investigators per the AFIT RTI protocol. The signed Informed Consent Document (ICD) of all human subjects are approved approved by the Air Force Research Laboratory (AFRL) Institutional Review Board (IRB). The signed ICD is available and all human subjects voluntarily participated in the data collection. All human targets were localized in the upright position. Although the height of each target varied, the heights were not taken into consideration as the sensors were placed at a height that would be obstructed by targets of various heights.

3.3 Assumptions

The following are the assumptions that were made in this research:

1. $P_l \sim \mathcal{N}\left(\bar{P}(d_l), \sigma^2\right)$
2. $\mathbf{n} \sim \mathcal{N}\left(\mathbf{0}, \sigma_n^2 \mathbf{I}_M\right)$
3. $\mathbf{y}|\mathbf{x} \sim \mathcal{N}\left(\mathbf{W}\mathbf{x}, \sigma_n^2 \mathbf{I}_M\right)$
4. $\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{C}_x\right)$

5. Calibration for all data collection was completed for at least 30 sets of **y** observations and is available.
6. Measurement noise and static losses are averaged out through the use of the mean of the calibration data.
7. All radios were oriented in the vertical position with the USB interface facing down.
8. All human targets are tracked in the upright position.
9. The number of targets is known.
10. Tracking filters for tracking or noise reduction are not used in this research.
11. Fade loss as a result of multipath is not significant enough to to be incorporated in the weighting model or regularization method.
12. Obstructions in the network affect signal propagation between links as a result of shadowing loss.
13. The pixel intensity of each pixel in the estimated scene **x** is constant through the area of the pixel.

3.4 System Models

This research applied MRTI from section 2.6.1, where $\mathbf{y}_{mean} = [\Delta\bar{r}_{1,n}, \Delta\bar{r}_{2,n}, ..., \Delta\bar{r}_{M,n}]^T$. Therefore, **y** is computed by $\mathbf{y} = \mathbf{y}_{mean} - \mathbf{y}_c$. The linear system model is defined by (2.4). A weight model and estimator needs to be chosen to estimate **x**.

Weight Model. The *Line Model* was chosen for the weight model, **W** from Section 2.3.2. Although the *NeSh Normalized Ellipse* model from Section 2.3.1 is the more popular weighting model in literature, the *Line Model* was chosen due to its lower complexity. Since the localization algorithm that is applied in this research adds additional complexity, cutting down on complexity can be beneficial for real-time applications. This model assigns a weight dependent on the path lengths of the links passing through the obstruction.

Regularization. *Tikhonov Regularization* from Section 2.4.1 was used to estimate **x** in the least-squares sense. The first order difference operator **Q** is discussed in Section 2.4.1

and the tunable regularization parameter α was chosen after data collection. As discussed in [1], the optimal α is dependent on the network setup and pixel size. The results of the regularization is presented in Chapter 4.

3.5 Localization Method

The focus of this research utilizes the K-means algorithm to cluster together higher intensity pixels to estimate the position of targets. For single target applications, the most common form of localization is taking the maximum pixel value from the estimated image scene $\hat{\mathbf{x}}$ as discussed in Section 2.7. However, for MTT, localization is more difficult. When multiple targets are present, there should be multiple areas where the pixel intensity would be higher in the areas in which targets are present.

K-means Clustering. Since the primary focus of this research is multiple target localization, K-means clustering from Section 2.7 will be utilized to estimate the location of multiple targets. When multiple targets are present in an image scene, the higher intensity pixels can be clustered together and localized using K-means. The associated squared error cost function for K-means is

$$J = \sum_{i=1}^{K} \sum_{x_j \in S_i} \left\| x_j - C_i \right\|^2, \tag{3.2}$$

where C_i is the cluster position of cluster k and x_j is the is the jth element of the pixels above a set threshold to be assigned to a cluster. The number of clusters K and the pixel locations to be clustered, x_j are the inputs to the K-means algorithm. Since it is assumed the number of targets is known, *apriori*, K can be appropriately chosen for the number of targets known to be present inside the network.

Pixel Intensity Threshold. Since the desired pixels are those that have a higher intensity, some type of threshold is warranted to segregate the pixels that have no targets present from the pixels that are occupied by targets. From experimental data, it was found that the statistics on the image $\mathbf{x}$ change with the regularization parameter α, pixel size Δ_p, the number of targets, and where in the network the targets are. Therefore, the threshold has to be robust enough to accommodate the change in intensity values between frames and flexible to handle the change of chosen parameters. The threshold T_c is used to determine which pixels are clustered into the K-means algorithm. The variance is first found from the estimated image scene which is

$$\sigma_n^2 = VAR[\hat{\mathbf{x}}_{Tik,n}], \tag{3.3}$$

where $\hat{\mathbf{x}}_{Tik,n}$ is the Tikhonov estimation at each frame n. The image scene can be modeled as Gaussian [1]. Pixels that are occupied by targets can be assumed to be much greater than the pixels unoccupied by targets. The threshold T_c mainly used for this research is

$$T_c = 3\sigma_n, \tag{3.4}$$

where $3\sigma_n$ is three times the standard variation at each frame n.

In summary:

- System Model: $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{n}$
- Measurement Model: $\mathbf{y} = [\Delta r_1, \Delta r_2, \cdots, \Delta r_M]^T$
- Calibration: $\mathbf{y}_c = [\bar{r}_{c,1}, \bar{r}_{c,2}, \cdots, \bar{r}_{c,M}]^T$
- Weight Model: $[\mathbf{W}]_{l,p}^{Line} = L_{l,p}\begin{cases} 1 & \text{if link } l \text{ traverses voxel } p \\ 0 & \text{otherwise} \end{cases}$
- Estimator: $\hat{\mathbf{x}}_{TIK} = \underset{x}{\operatorname{argmin}} \left(||\mathbf{W}\mathbf{x} - \mathbf{y}||^2 + \alpha ||\mathbf{Q}\mathbf{x}||^2 \right)$
- Tikhonov Matrix: $\mathbf{Q} \triangleq \mathbf{D}_x^T\mathbf{D}_x + \mathbf{D}_y^T\mathbf{D}_y + \mathbf{D}_z^T\mathbf{D}_z$
- Pixel Threshold: $T_C = 3\sigma_n, \sigma_n = VAR[\hat{\mathbf{x}}_{Tik,n}]$
- Localization: $J = \sum_{i=1}^{K} \sum_{x_j \in S_i} \left\| x_j - C_i \right\|^2$

3.6 Choosing Model and Experiment Parameters

Trade-off analysis has been conducted in [56] and [1] for model parameters. However, model parameters α and Δ_p based off of review from preliminary results. Data from these experiments were analyzed using a range of values for α and Δ_p and the resulting Root Mean Squared Error (RMSE) for each data set was compared.

3.7 Simulated Truth Data

The *Cylindrical Human model* described in Section 2.5 was used to simulate all the stationary truth images. $\mathbf{C}_h$ is set to be the known (x, y) coordinates of the targets to be localized. The model can mathematically described as

$$[\mathbf{x}]_{CHM} = \begin{cases} 1, & \text{if } ||(x, y)_p - \mathbf{C}_h|| < R_h \\ 0, & \text{otherwise} \end{cases}, \tag{3.5}$$

where $\mathbf{x}_{CHM}$ is a $[L_x, L_y]$ matrix set by the pixel size Δ_p, $(x, y)_p$ is the center coordinate of each pixel p, and R_h is the human radius. Therefore, the true attenuation image model $\mathbf{x}_{CHM}$ contains a 1 in the pixel location that is centered on $\mathbf{C}_h$ contained within R_H and zeros elsewhere. The simulated $\mathbf{y}_{sim}$ data is calculated by the linear model (2.4) from Section 2.2.4. The vector $\mathbf{y}_{sim}$ can mathematically be shown as

$$\mathbf{y}_{sim}{=}\mathbf{W}_{line}\mathbf{x}_{CHM}{+}\mathbf{n}_{sim}, \tag{3.6}$$

where $\mathbf{n}_{sim}$ is a simulated AWGN vector of variance σ^2_{sim}. Using $\mathbf{y}_{sim}$ from (3.6), the simulated image scene using *Tikhonov Regularization* is

$$\hat{\mathbf{x}}_{sim} = \left(\mathbf{W}^T_{line}\mathbf{W}_{line} + \alpha\mathbf{Q}\right)^{-1} \mathbf{W}^T_{line}\mathbf{y}_{sim}, \tag{3.7}$$

where $\mathbf{Q}$ and the line weighting matrix $\mathbf{W}_{line}$ are defined in Section 3.4. Substituting the *Tikhonov* matrix from Section 2.4.1 yields

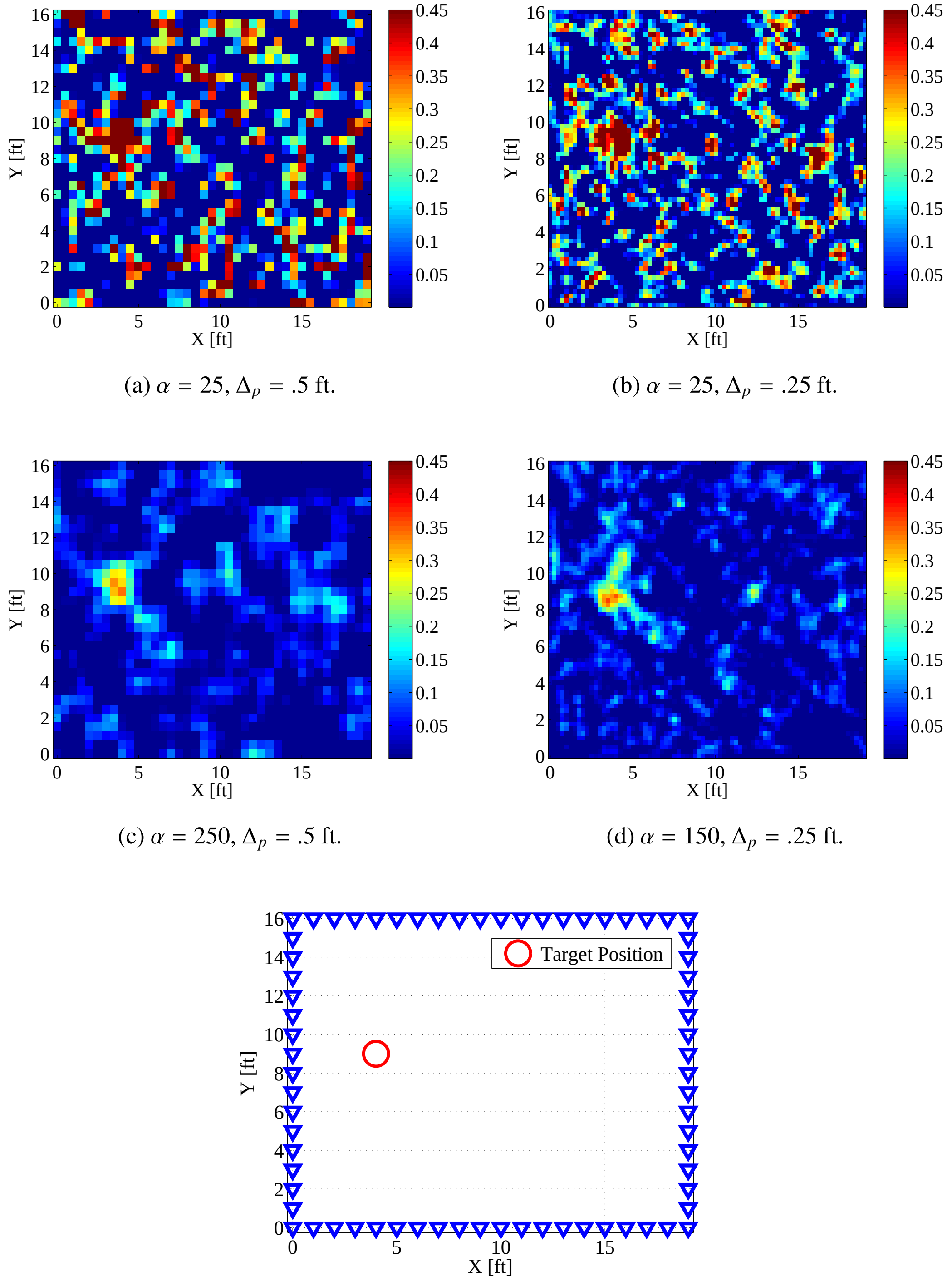
$$\hat{\mathbf{x}}_{sim} = \mathbf{\Pi}_{Tik}\mathbf{y}_{sim}, \tag{3.8}$$

where $\mathbf{\Pi}_{sim}$ can be computed in advance for both simulation and real-time applications.

Table 3.2: Simulated truth data parameters.

Parameter	**Value**	**Description**
σ_N^2	10	Noise variance (dB^2)
R_h	1.1	Human radius for cylindrical model (ft)
Δ_p	0.25 & 0.5	Pixel width (ft)

Table 3.2 shows the parameters used in the simulated truth data. As shown in Figures 3.5-3.7 with almost no regularization, where $\alpha = 25$, the noise in the network is more apparent in the estimated image scene. As discussed in Section 2.4.1, the purpose of α is to suppress the noise spikes in the image. In [1], α was varied for the given network until the MSE was minimized. In this research, α was increased until a visually acceptable image scene was found. The advantage of α is that it can be changed by the user in both real-time and when analyzing the data post collection. As seen in Figures 3.5-3.7, the desired α is changed with pixel size. As the pixel size grows smaller, the energy is spread throughout a higher number of pixels and thus the properties of the image scene changes. For a pixel size of $\Delta_p = 0.5$ ft, the optimal regularization parameter was found to be $\alpha = 250$. However, when $\Delta_p = 0.25$ ft, the optimal regularization parameter was found to be $\alpha = 150$.



(a) α = 25, Δ_p = .5 ft.

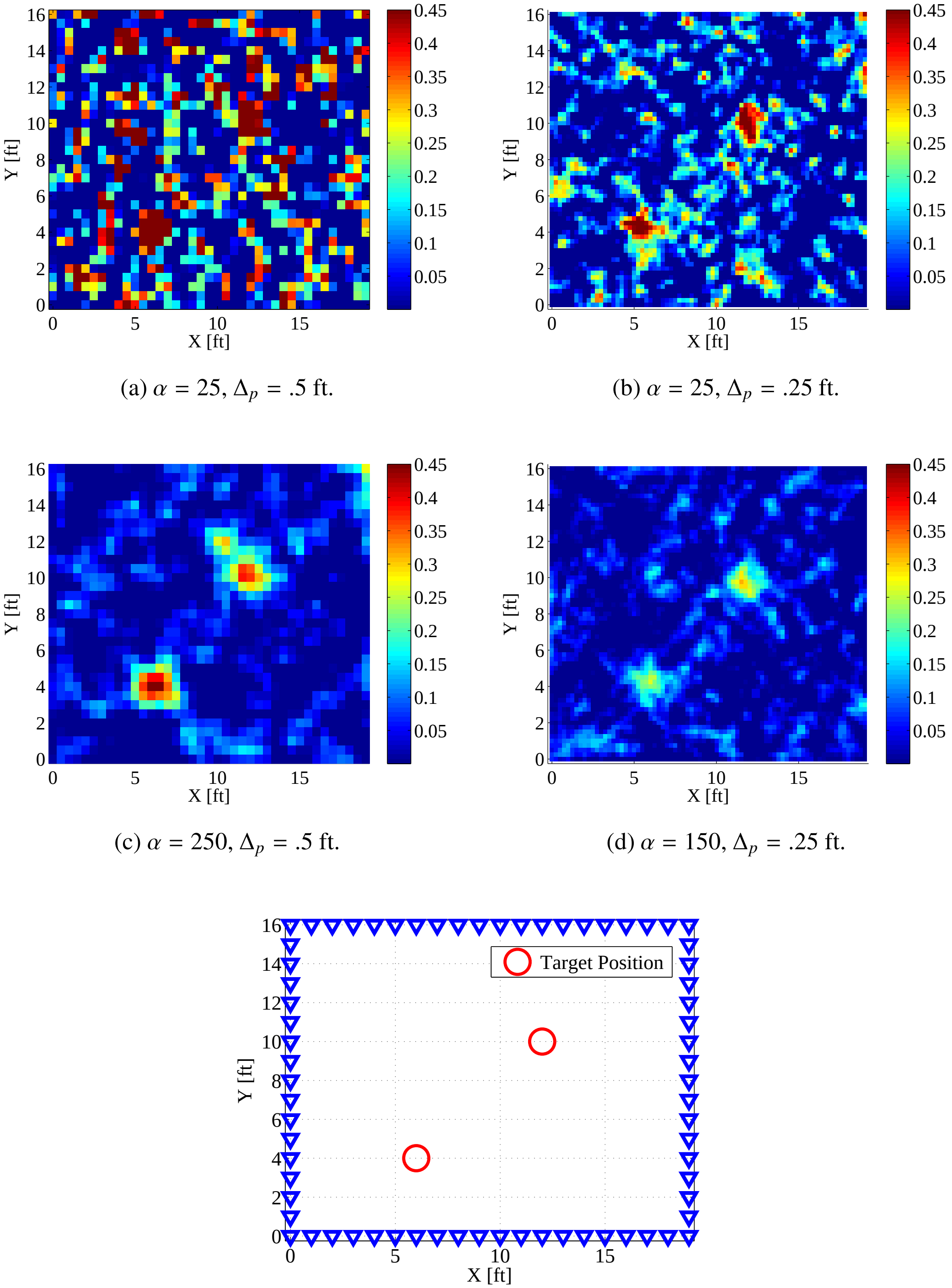
(b) α = 25, Δ_p = .25 ft.

(c) α = 250, Δ_p = .5 ft.

(d) α = 150, Δ_p = .25 ft.

(e) Single target at (4, 9) ft.

Figure 3.5: Truth Images: Single target at (4, 9) ft.



(a) $\alpha = 25$, $\Delta_p = .5$ ft.

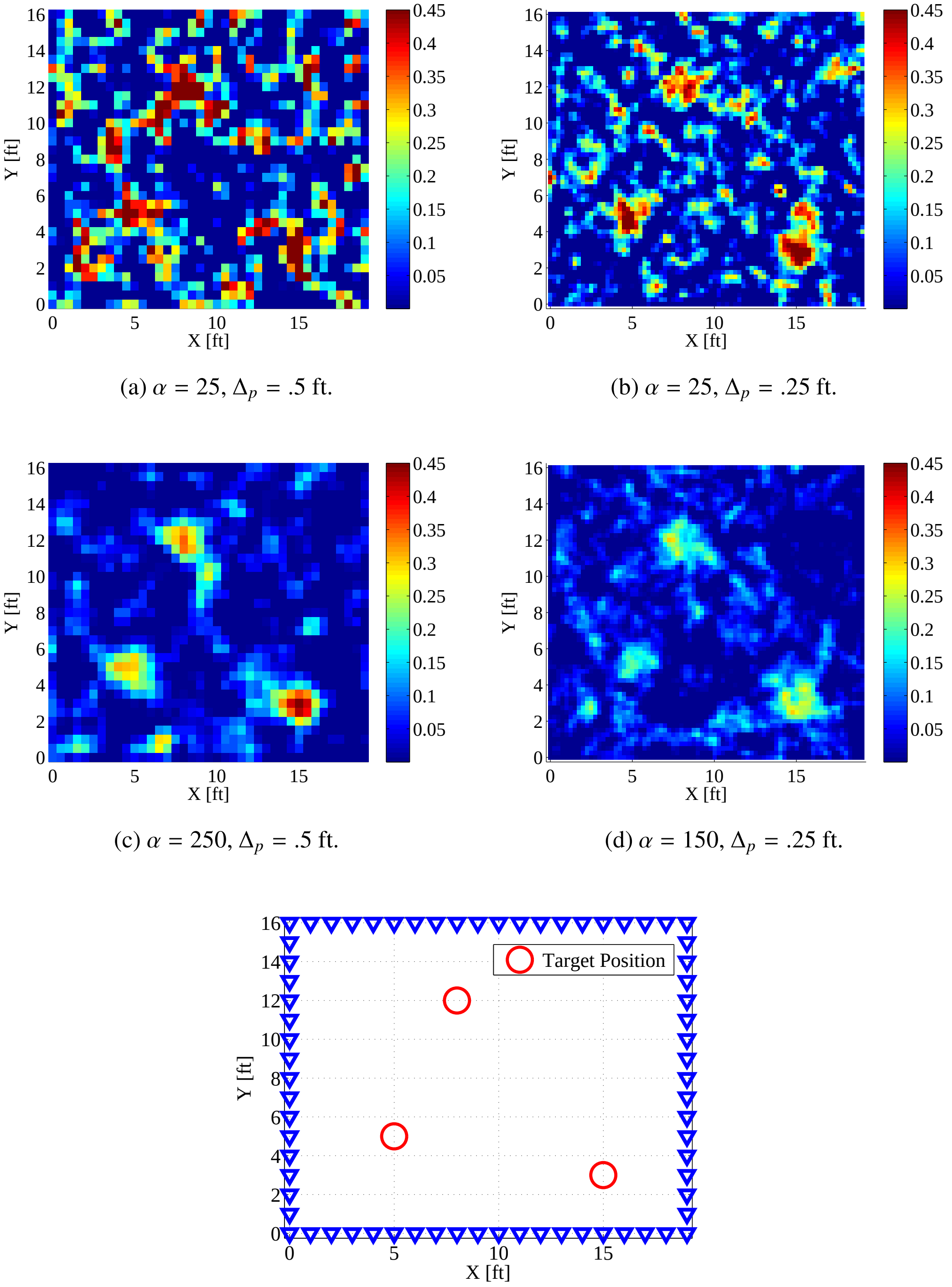
(b) $\alpha = 25$, $\Delta_p = .25$ ft.

(c) $\alpha = 250$, $\Delta_p = .5$ ft.

(d) $\alpha = 150$, $\Delta_p = .25$ ft.

(e) Two targets at $(6, 4)$ and $(12, 10)$ ft.

Figure 3.6: Truth Images: Two targets at $(6, 4)$ and $(12, 10)$ ft.



(a) $\alpha = 25$, $\Delta_p = .5$ ft.

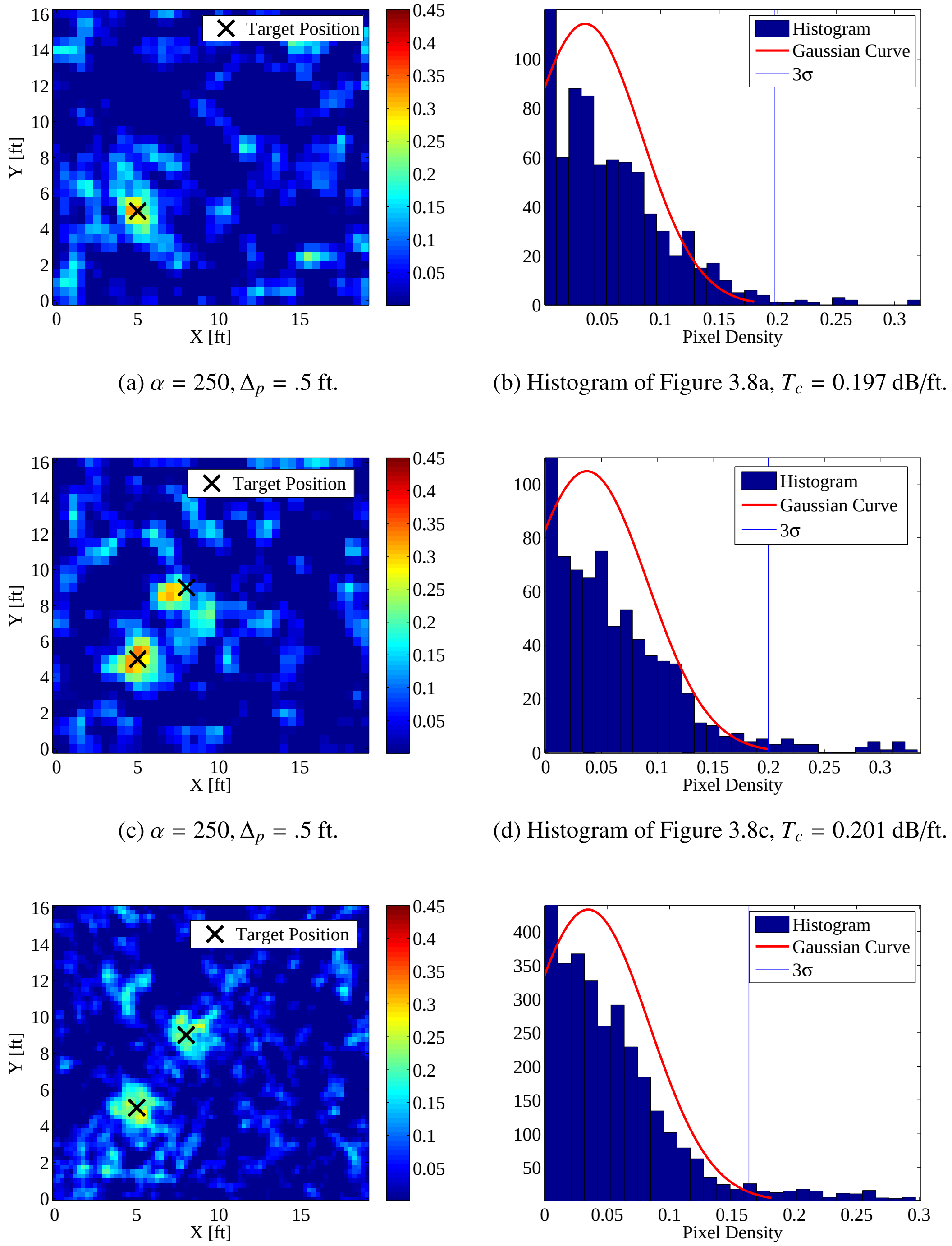
(b) $\alpha = 25$, $\Delta_p = .25$ ft.

(c) $\alpha = 250$, $\Delta_p = .5$ ft.

(d) $\alpha = 150$, $\Delta_p = .25$ ft.

(e) Three targets at $(5, 5)$, $(8, 12)$ and $(15, 3)$ ft.

Figure 3.7: Truth Images: Three targets at $(5, 5)$, $(8, 12)$ and $(15, 3)$ ft.



(a) α = 250, Δ_p = .5 ft.

(b) Histogram of Figure 3.8a, T_c = 0.197 dB/ft.

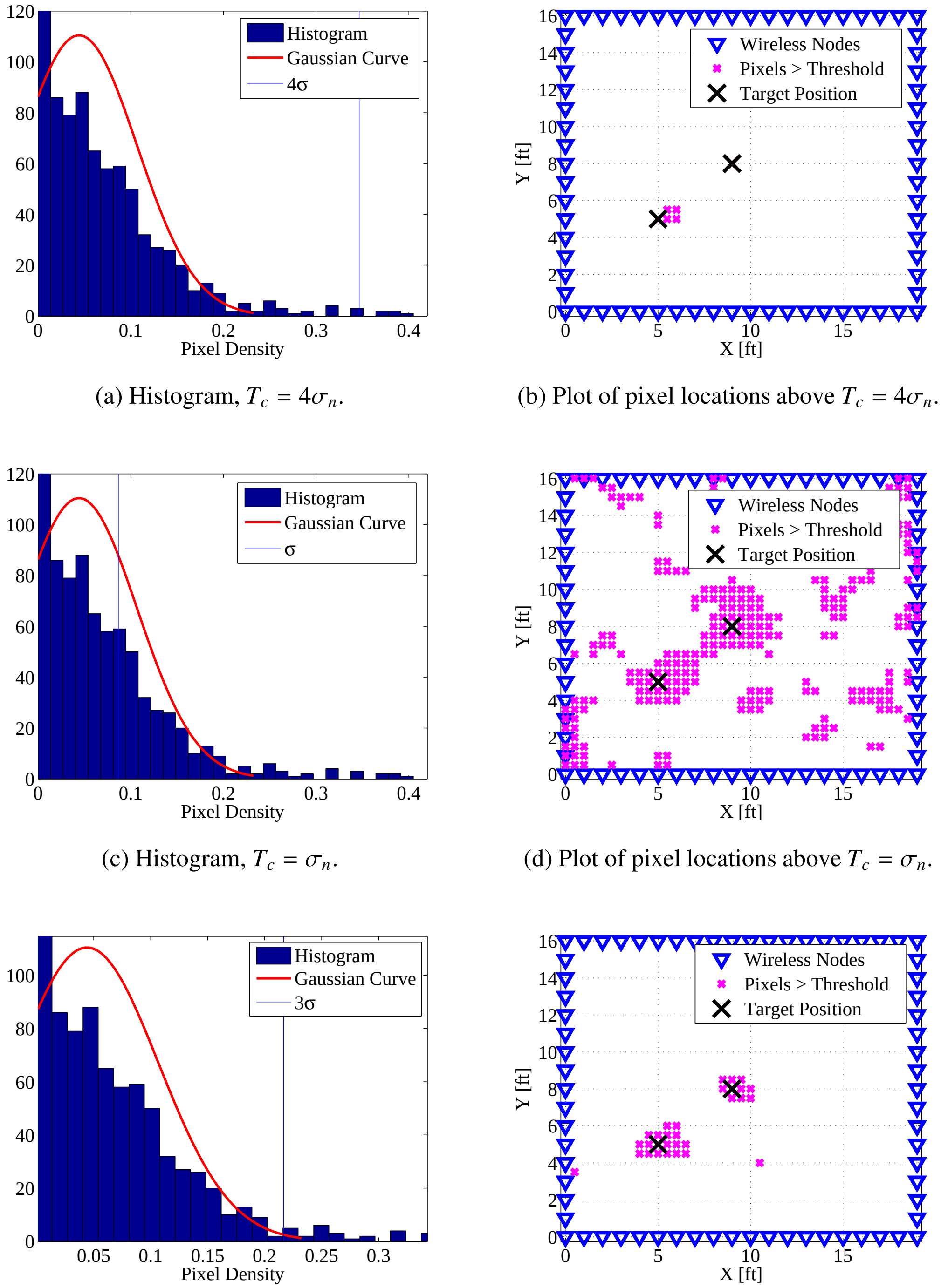
(c) α = 250, Δ_p = .5 ft.

(d) Histogram of Figure 3.8c, T_c = 0.201 dB/ft.

(e) α = 150, Δ_p = .25 ft.

(f) Histogram of Figure 3.8e, T_c = 0.164 dB/ft.

Figure 3.8: Histograms of frames with varying targets and parameters.



(a) Histogram, $T_c = 4\sigma_n$.

(b) Plot of pixel locations above $T_c = 4\sigma_n$.

(c) Histogram, $T_c = \sigma_n$.

(d) Plot of pixel locations above $T_c = \sigma_n$.

(e) Histogram, $T_c = 3\sigma_n$.

(f) Plot of pixel locations above $T_c = 3\sigma_n$.

Figure 3.9: Histogram of Fig. 3.8c with varying threshold values T_c. Pixel locations of the pixels above the threshold are plotted to the right of the histograms.

3.7.1 Cluster Threshold.

Fig. 3.8 shows three different histograms for three frames of truth images. The frames vary in target location as well model parameters, Δ_p and α. The histograms are formed with 30 bins and are fitted to a Gaussian curve. In Fig. 3.8d, the threshold using (3.4) is $T_c = 0.201$ dB/ft. In Fig. 3.8f, the targets are in the same location as in Fig. 3.8d, but the parameters Δ_p and α are different. Subsequently, the statistics shown in the histogram are different. Therefore, it makes statistical sense that the threshold for Fig. 3.8f has changed to $T_c = 0.164$ dB/ft. The threshold value, T_c needs to be computed before each frame since the statistics of the image scene $\hat{\mathbf{x}}$ can change from frame to frame.

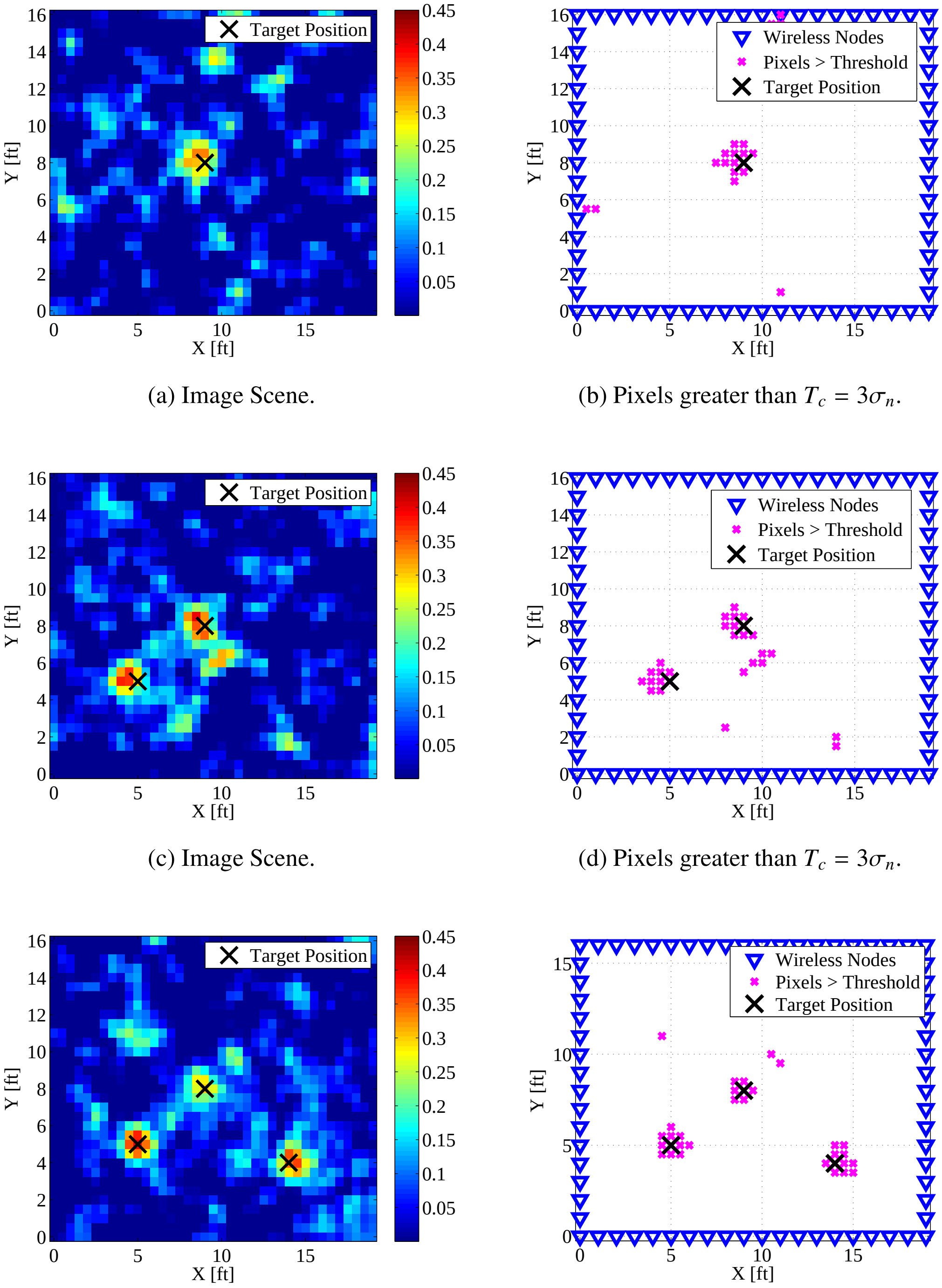
Fig. 3.9 illustrates the outcomes after changing threshold for the same image frame. In Fig. 3.9c, when $T_c = 4\sigma_n$, the threshold is set too high, where an insufficient amount of pixel densities are above the threshold in Fig. 3.9d. In Fig. 3.9a, when $T_c = 4\sigma_n$, the threshold is set too low, where a higher than necessary amount of pixel densities are above the threshold in Fig. 3.9b. If the threshold is not stringent enough, lower pixel densities can cause the K-means clustering process to cluster together insignificant pixels. As shown in Fig. 3.9e, the essential pixel densities make make it past the threshold when $T_c = 3\sigma_n$.

3.7.2 Application of K-means Clustering.

Once the pixels above the threshold are identified, the locations of those pixels are inputted into the K-means algorithm from (3.2). Fig. 3.10 illustrates a set of images of frames with various targets in the left column. In the right column are plots of the pixel locations with densities above $T_c = 3\sigma$. Given the *prior* information, K, the (x, y) locations are assigned to a cluster and the centroids represent the localization estimate of each target.

First Iteration of K-means Clustering. Figs. 3.11a, 3.12a, and 3.13a represent the first iteration of the K-means algorithm. Since all pixels above the threshold T_c are assigned to a cluster, there is a chance of outliers being above the threshold. Erroneous pixel density spikes can occur in the image estimation. Fig. 3.11a, 3.12a, and 3.13a show the centroid positions of all the pixels that are above the threshold, T_c. The figures show that there are pixels that are segregated from the denser clusters which are around the targets. The pixels that are not close to a larger group of pixels can subsequently affect the centroid positions. Using a cluster radius of $R_c = 3.25$ ft would minimize the effects of segregated higher pixel values. Any pixels that lay outside the given radius from the initial cluster centroids can be deleted from the centroid calculation.

Second Iteration of K-means Clustering. Figs. 3.11b, 3.12b, and 3.13b illustrates the second iteration of K-means after pixels outside of the R_c radius are deleted from the K-means algorithm. The clusters which contained pixels with a greater number of outliers had initial centroids that were further away from the denser pixel clusters. After the outliers were deleted, the centroids moved closer to the groups of pixels closer to the target. In Fig. 3.11, the RMSE was 1.71 ft. After the pixels outside the $R_c = 3.25$ were ignored and the second K-means algorithm was run, the RMSE improved to 0.45 ft. Similarly, in Fig. 3.12, the RMSE improved from 0.90 ft to 0.38 ft. In Fig. 3.13, the RMSE improved from 1.16 ft to 0.45 ft. However, it is important to note that in these cases, the RMSE error improved because the pixel outliers were pulling the centroid locations away from the true position of the targets. In these cases, the denser groups of pixels were closer to the true target positions. Conversely, if the denser groups of pixels are not near the true target locations, a second iteration of K-means clustering may or may not provide an improved localization estimate.



(a) Image Scene.

(b) Pixels greater than $T_c = 3\sigma_n$.

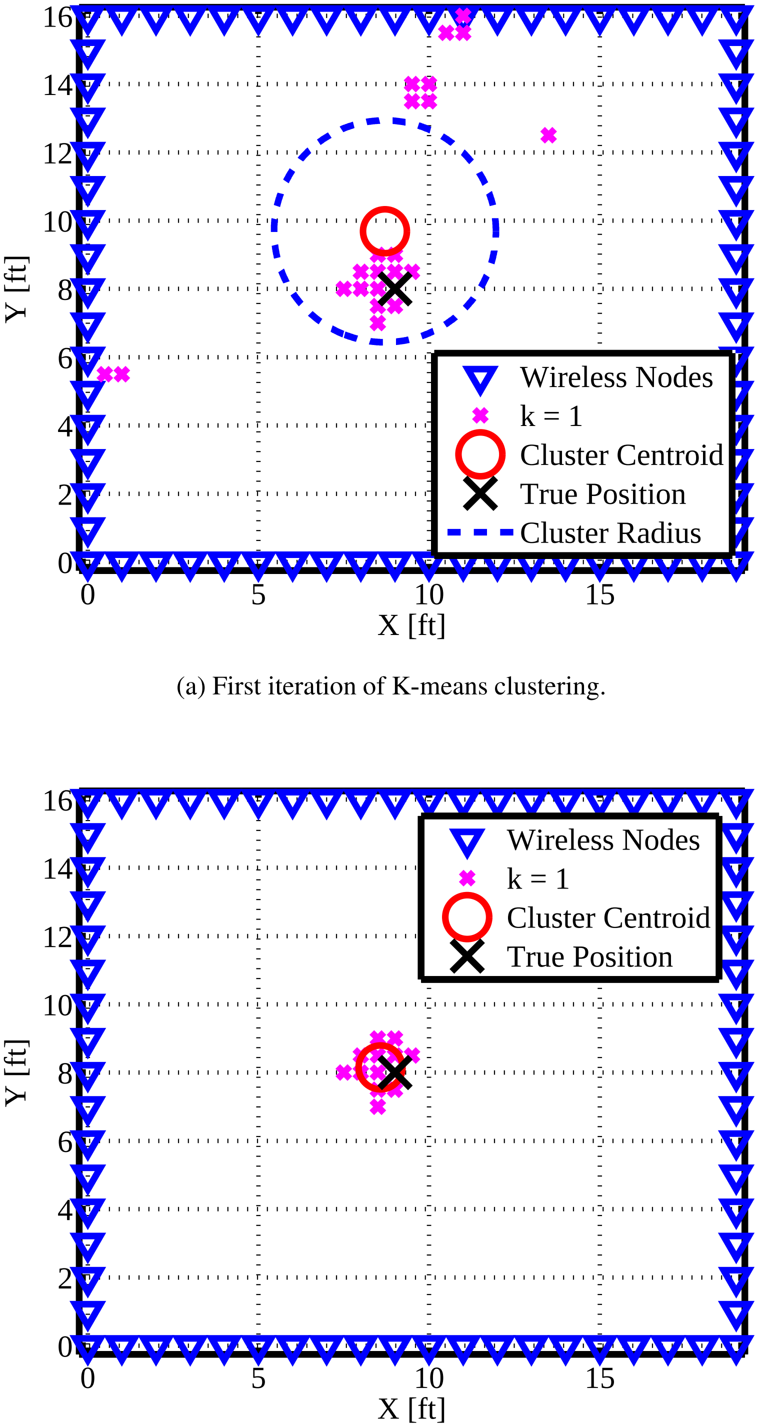
(c) Image Scene.

(d) Pixels greater than $T_c = 3\sigma_n$.

(e) Image Scene.

(f) Pixels greater than $T_c = 3\sigma_n$.

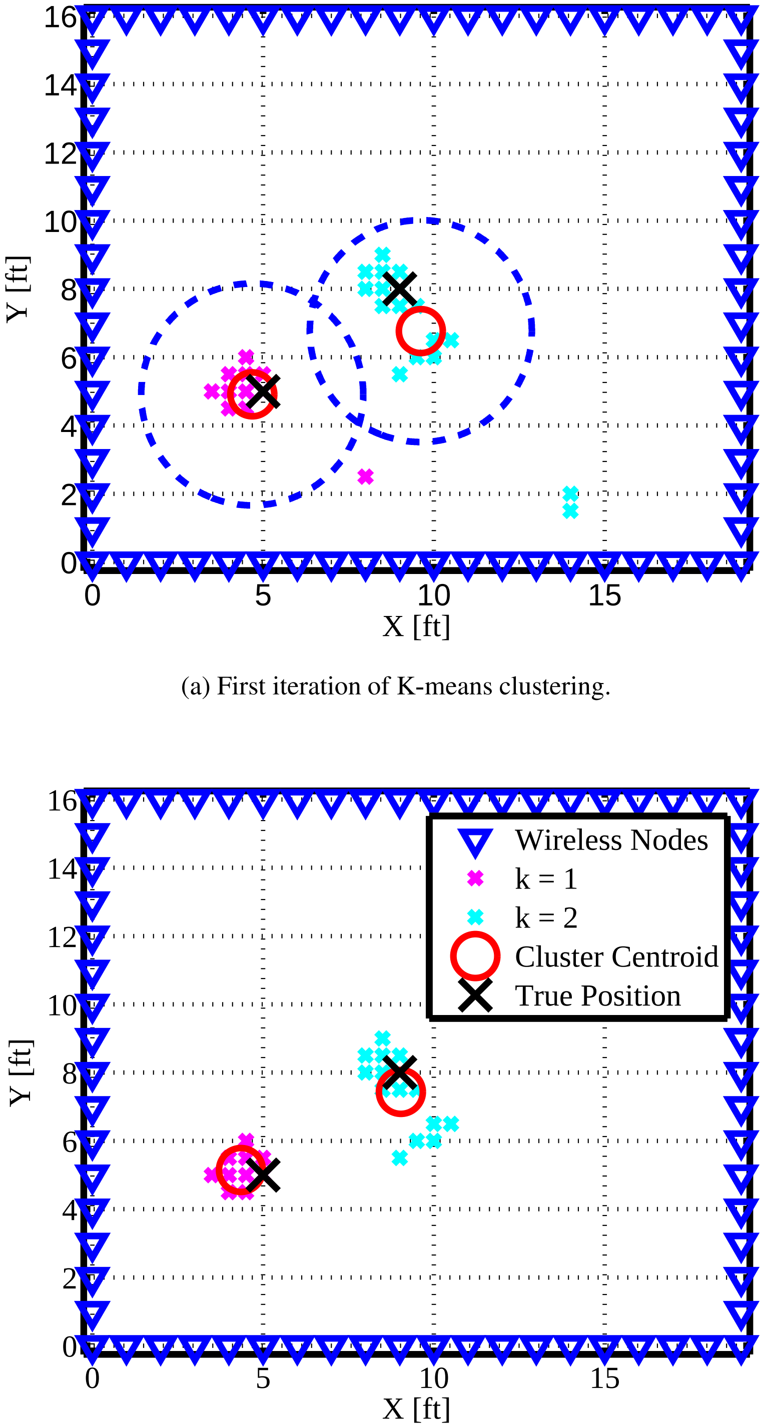
Figure 3.10: Image scenes and pixel threshold locations with multiple targets.



(a) First iteration of K-means clustering.

(b) Second iteration of K-means clustering.

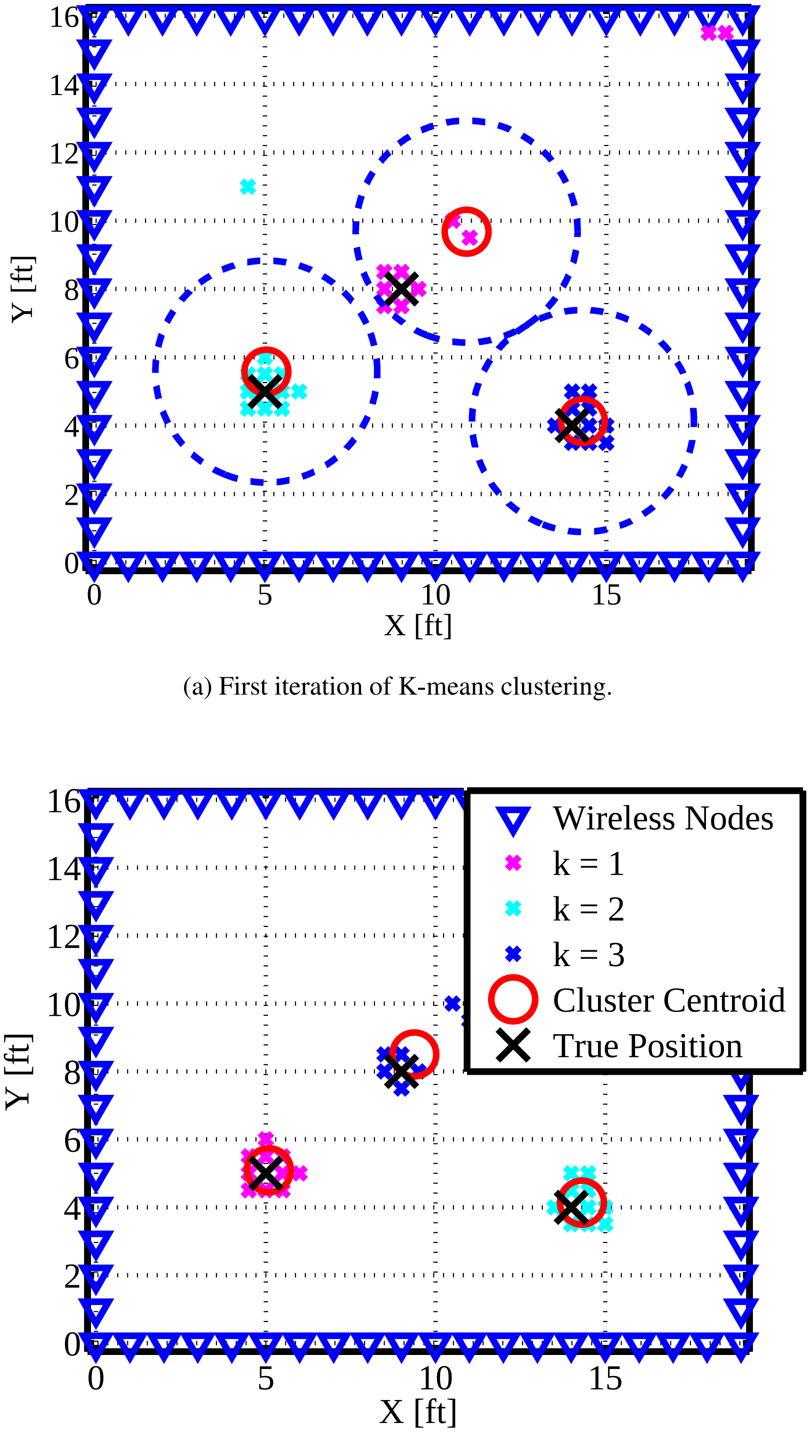
Figure 3.11: K-means Localization: target at (9, 8) ft.



(a) First iteration of K-means clustering.

(b) Second iteration of K-means clustering.

Figure 3.12: K-means Localization: targets at $(5, 5)$ and $(9, 8)$ ft.



(a) First iteration of K-means clustering.

(b) Second iteration of K-means clustering.

Figure 3.13: K-means Localization: targets at $(5,5)$, $(9,8)$, and $(14,4)$ ft.

3.8 Experiment Design

This research focuses on a variation of experiments with the designed WSN discussed in Section 3.2. Single target stationary and tracking localization is used to compare K-means clustering with MAP localization. Since MTT is more complex and cannot use MAP localization, classification of K-means clustering will be the only focus of multiple stationary and moving targets. With all experiments, parameters used to generate the results are defined and recorded. Network calibration was completed prior to each experiment recording, which is considered the baseline for the network without any targets present. This baseline can include obstructions that would essentially be calibrated out as the obstructions are not considered targets of interest in this research. For all tracking data, a metronome was used synchronize movement and the rate at which the frames are recorded.

3.9 Data Analysis

All experiment data and processing analysis was accomplished using `MATLAB`®. Although, preliminary data can be viewed in near real-time through the GUI, data processing for all experiments were completed post data collection. All analysis for single and multiple targets were completed using the same process with one exception. Single target K-means localization is able to be compared to MAP localization while multiple targets are only tracked using K-means clustering.

3.9.1 Experimental Challenges.

There are some experimental challenges that need to be overcome when utilizing the RTI motes. During data collection, the motes can give Not a Number (NaN) readings for various RSS links in the $\mathbf{y}$ data. Steps must be accomplished to successfully solve for the image scene of a frame.

Hardware Challenges. Upon investigation, it was found that the the number of links to have a NaN in any particular frame were less than 7 percent of the total RSS vector. However, there are frames that did not have any NaN readings, but when there were NaNs, in order to solve for the image scene, $\hat{\mathbf{x}}_{Tik}$, the NaN data must be removed from the **y** vector. Let the number of links in **y** be then number of nominal links minus the number of NaN links, L_{NaN}. The new number of links in **y** would be

$$L' = L - L_{NaN}, \tag{3.9}$$

where the vector of RSS links would now be of length $[L' \times 1]$. In order to accomplish this, the location of the NaN readings must be deleted from the corresponding location in the vector **y**. To successfully solve for $\hat{\mathbf{x}}_{Tik}$, the corresponding rows of the NaN locations would need to be deleted from the weighting matrix, $\mathbf{W}_{Line}$. The new size of the weighting matrix would be $[L' \times P]$ rather than $[L \times P]$. By deleting the respective row location in $\mathbf{W}_{Line}$ containing NaNs, the modified weight matrix can be represented as $\mathbf{W}'$. Using *Tikhonov Regularization* as the chosen image estimator outlined in Section 3.4, the image estimate can mathematically be defined as

$$\hat{\mathbf{x}}_{Tik} = \left((\mathbf{W}')^T\mathbf{W}' + \alpha\mathbf{Q}\right)^{-1}\mathbf{W}'\mathbf{y}', \tag{3.10}$$

where

$$\mathbf{\Pi}'_{Tik} = \left((\mathbf{W}')^T\mathbf{W}' + \alpha\mathbf{Q}\right)^{-1}\mathbf{W}', \tag{3.11}$$

$$\hat{\mathbf{x}}_{Tik} = \mathbf{\Pi}'_{Tik}\mathbf{y}'. \tag{3.12}$$

Negative Pixel Density. Since the vector of **y** can contain negative RSS when measuring the differences in RSS links, it is possible for $\hat{\mathbf{x}}$ to contain negative pixel density values. In [42], Martin et al. describe a method to force an $\hat{\mathbf{x}}$ containing only positive values. The other alternative is to assume negative $\hat{\mathbf{x}}$ values are the same as being close to a value of 0 dB/ft. It is computationally cheap to set any negative $\hat{\mathbf{x}}$ entries to 0. Thus in this research, all negative $\hat{\mathbf{x}}$ entries will be set to 0.

3.9.2 Performance Metrics.

Estimated positions can be drawn from the $\hat{\mathbf{x}}$ data from the RTI network using the discussed localization methods. To evaluate the accuracy of the location estimate for all the targets, an accuracy metric can be used. The RMSE of the localized estimate is commonly used in the literature [2]. The RMSE of one or more targets targets can mathematically be computed as

$$\bar{e} = \left(\frac{1}{T} \frac{1}{N} \sum_{t=1}^{T} \sum_{n=1}^{N} \left(\hat{z}_t(n) - z_t(n) \right)^2 \right)^{1/2}, \tag{3.13}$$

where T is the number or targets, N is the number of frames, $\hat{z}_t(n)$ is the estimated position for target t at frame n, and $z_t(n)$ is the true position of target t at frame n.

3.10 Chapter Summary

This chapter described the tools and equipment used for all experiments completed in this research. The methodologies used to establish the network design and localization of targets are established in this chapter. Simulated truth data, baseline data collection, and the methods used to analyze data have established.

IV. Results and Discussion

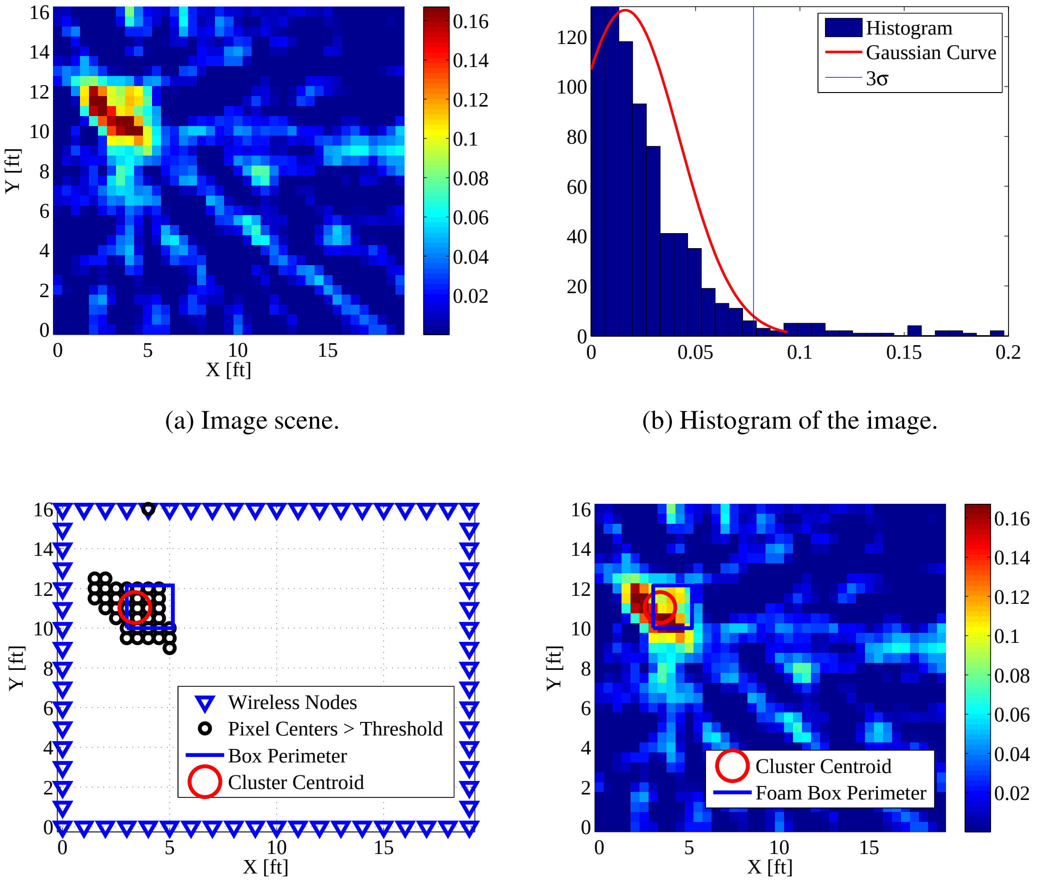
This chapter contains the results of stationary localization for single and multiple targets as well as motion tracking with both obstructed and unobstructed environments. The use of K-means clustering is utilized to localize the targets. The classification of the K-means clustering algorithm to geolocate multiple targets is discussed in this section. The focus will be to characterize the use of K-means for single and multiple targets localization. RF absorbing foam boxes are used as targets inside the designed RTI network for the experimental truth images. Stationary target localization experiments are with one, two, and three targets. The results of motion tracking with one and two targets are presented with and without obstructions. The results of this section will be discussed and classified using the performance metrics discussed in Section 3.9.2.

4.1 Experimental Truth Images

A series of experimental truth images using foam boxes were used to obtain a visual performance baseline of the designed RTI network described in Section 3.2. The dimensions of the foam box was such that it would be tall enough to be in the LOS of the sensor height with an overall dimension $[L \times W \times H]$ of approximately $[2.15 \times 2.15 \times 3.45]$ ft. The goal of the truth images is to gather a baseline performance of the RTI network by clustering the pixels above the threshold T_c using K-means after solving for the image scene $\hat{\mathbf{x}}$.

The foam boxes were moved to different parts of the network. Calibration was completed prior to placing the foam box target inside the network. As outlined in Section, 3.3, calibration was completed for at least 30 frames. In Fig. 4.1 and Fig. 4.2, the lower left corner of the box was placed at $(3, 10)$ and $(2, 2)$ respectively. All pixels with densities above the threshold $T_c = 3\sigma_n$ are plotted in Fig. 4.1c and Fig. 4.2c. Most of the pixel locations are contained inside the foam box perimeter resulting in the cluster centroids to be contained inside the box perimeter after performing one iteration of K-means clustering. The cluster variance for Fig. 4.1 and Fig. 4.2 was 13.6 and 6.3 ft^2 respectively.

In Fig. 4.3, the bottom left corner of the box was placed at $(8, 8)$. The pixels above the threshold $T_c = 3\sigma_n$ were further spread out than Fig. 4.1 and Fig. 4.2 resulting in a higher variance of 24.8 ft^2. However, K-means clustering found the centroid to be near the center of mass of the foam box. As described in Section 2.5, the CRLB derived in [1] showed that the CRLB is the lowest towards the middle of the network and higher near the corners. With the experimental images taken in this research, the opposite conclusion was formed. Images with targets near the corners of the network appeared to have denser clusters than when targets were near the center of the network.



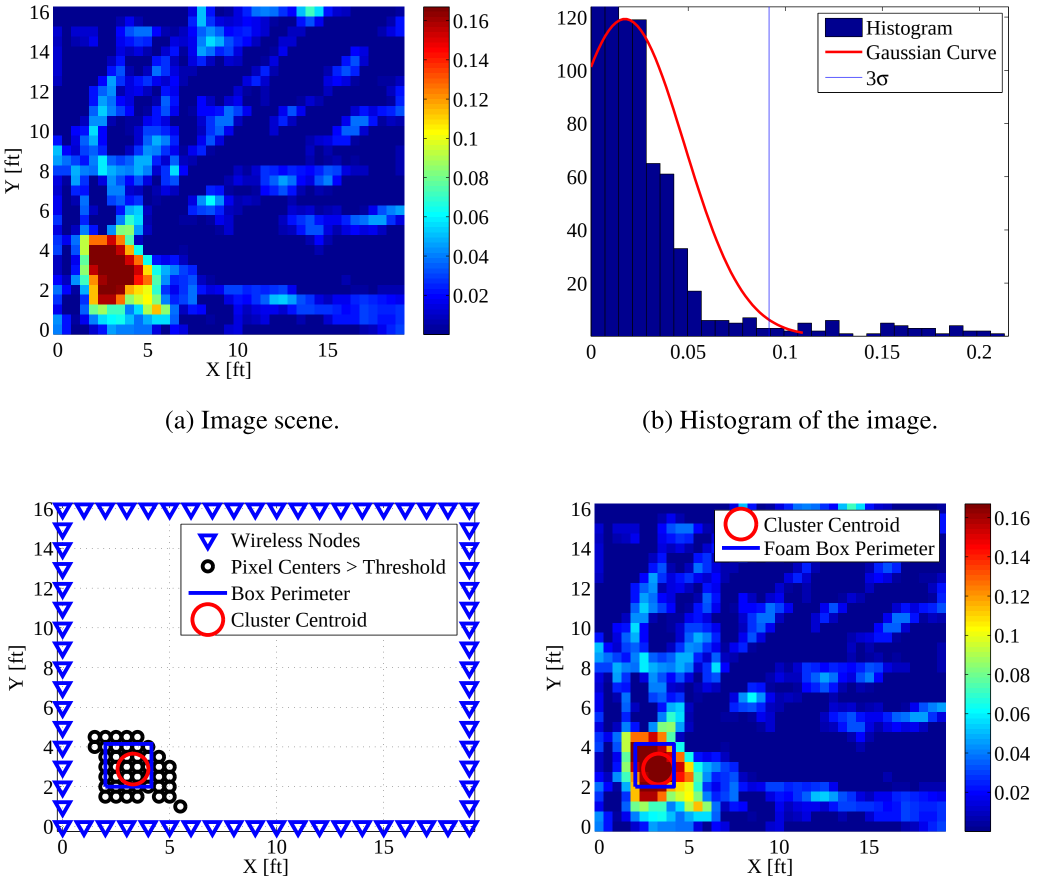
(a) Image scene.

(b) Histogram of the image.

(c) K-means clustering.

(d) Image scene after K-means clustering.

Figure 4.1: Truth image of foam box with the bottom left corner at (3, 10) ft with $\alpha = 250$, $\Delta_p = 0.5$ ft, and $T_c = 3\sigma_n$



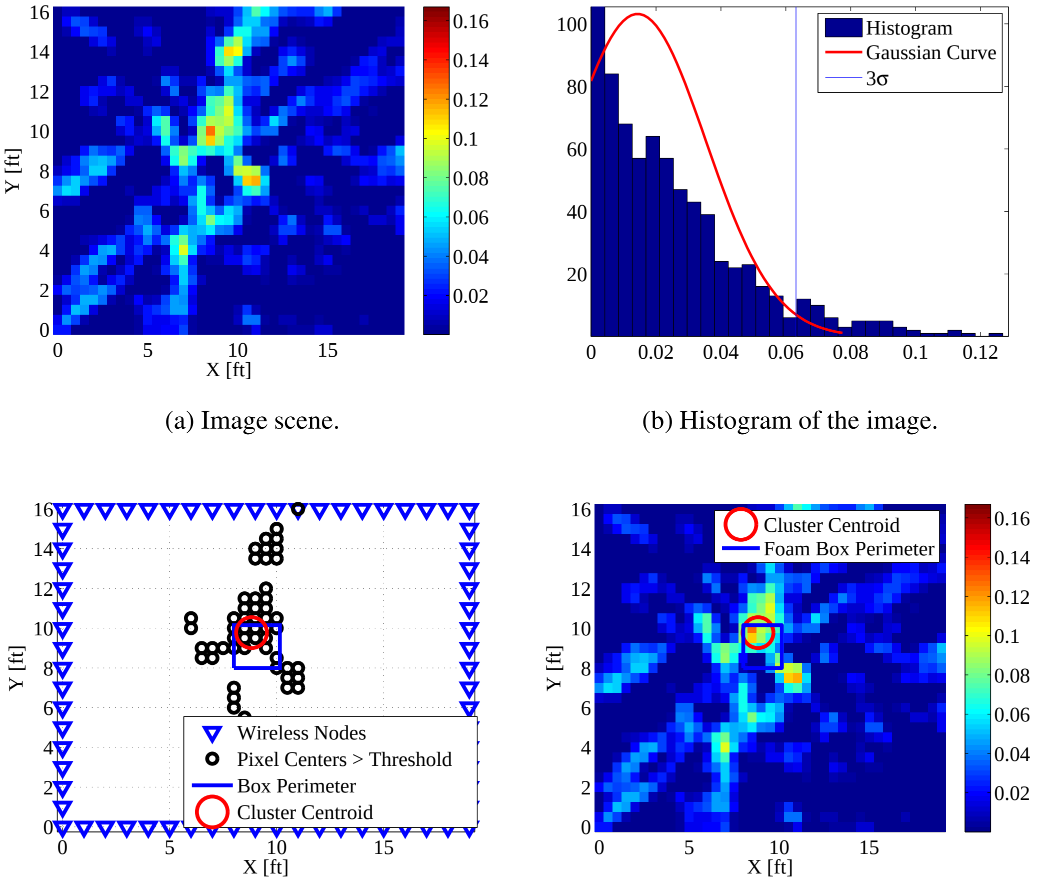
(a) Image scene.

(b) Histogram of the image.

(c) K-means clustering.

(d) Image scene after K-means clustering.

Figure 4.2: Truth image of foam box with the bottom left corner at (2, 2) ft with $\alpha = 250$, $\Delta_p = 0.5$ ft, and $T_c = 3\sigma_n$



(a) Image scene.

(b) Histogram of the image.

(c) K-means clustering.

(d) Image scene after K-means clustering.

Figure 4.3: Truth image of foam box with the bottom left corner at (8, 8) ft with $\alpha = 250$, $\Delta_p = 0.5$ ft, and $T_c = 3\sigma_n$

4.2 Stationary Target Localization

Multiple localization experiments were done with one, two, and three human targets. K-means localization is used to geolocate the position of the target(s) inside the network. With one target, K-means clustering can be compared with MAP localization where the pixel with the highest density is chosen to be estimated target position. In all experimental localization, one frame based on one observation of $\mathbf{y}$ is used to localize the target(s) inside the network. The purpose of this section is to characterize the results of K-means clustering. The K-means clustering process used to localize the targets is outlined in this section. All localization is completed with 2 iterations of K-means clustering.

The objective of this section is to determine in which cases one iteration of K-means would be sufficient and in which cases 2 iterations would be beneficial. This section examines how noise or outliers can affect the results of K-means clustering. This section will examine the results of changing the parameters such as the pixel threshold and pixel size.

4.2.1 Single Target Stationary Localization.

In Fig. 4.4, there is a human target at $(5, 5)$ ft. After the pixel locations above the threshold $T_c = 3\sigma_n$ are kept, the first K-means iteration is performed. The error for the frame was $\bar{e} = 1.43$ ft after the first iteration of K-means clustering. After the second iteration of K-means clustering, $\bar{e} = 0.38$ ft. In this situation, there were isolated pixel values over 10 ft from the target position that were above the pixel density threshold. This caused the centroid center to be biased. Therefore, since the second K-means iterations discarded the pixel locations outside the $R_c = 3.25$ ft radius, the isolated pixels did not affect the new cluster centroid location. The final estimate for the target was closer to the dense group of pixels around the true target location.

In Fig. 4.6, a comparison is made between MAP and K-means localization for a single target at $(9, 8)$ ft. This location was chosen for this comparison due to the pixel density of the image being spread out when a target is closer to the middle of the network. In this particular scenario, the maximum pixel density is 3.54 ft away from the true position. After the K-means clustering localization, the localization error for the frame is $\bar{e} = 0.69$ ft. In this case, K-means was more accurate because the highest pixel density value was more than further away from the target position than the K-means cluster. The cluster of pixels above the threshold congregated around the target causing the attenuation shown in the image scene. As a result, the K-means clustering localization computed a centroid near the target position.

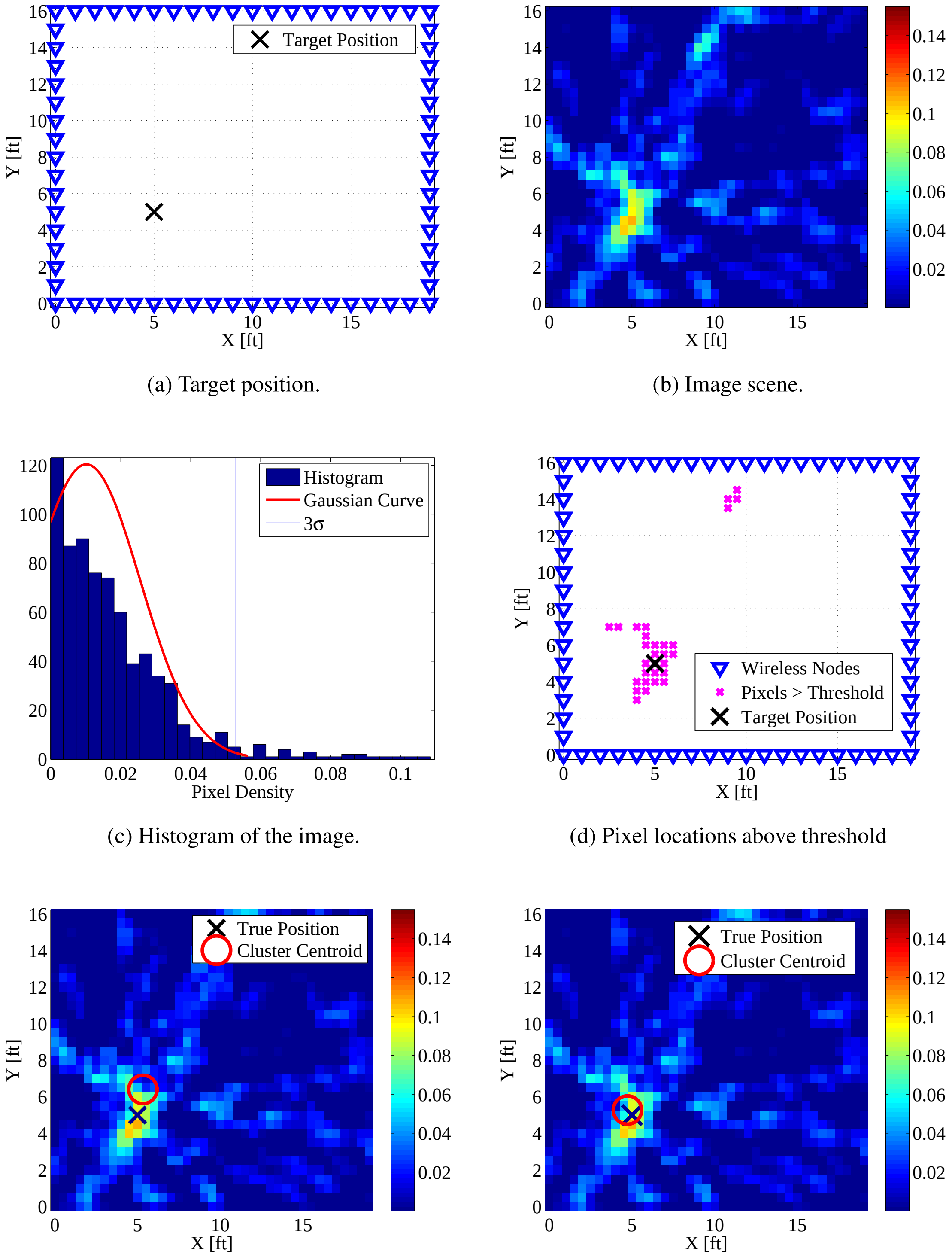
4.2.2 Multiple Target Stationary Localization.

In Fig. 4.7, 2 targets are at $(3, 11)$ and $(12, 4)$ ft. The error for one frame after both K-means iterations was $\bar{e} = 0.78$ ft. There was no change in cluster centroid positions because all the clustered pixels were inside the $R_c = 3.25$ ft radius after the first K-means clustering iteration. Therefore, there was no change to the RMSE after the second iteration.

In Fig. 4.11, with 3 targets at $(2, 2)$, $(5, 11)$, and $(17, 14)$ ft, the the error was $\bar{e} = 0.28$. Similar to Fig. 4.7, the RMSE did not change because the pixels above the threshold were inside the $R_c = 3.25$ ft radius after this first iteration of K-means clustering. In Fig. 4.11 the initial K-means iteration picked a centroid containing pixels segregated outside the radius of the centroid near $(17, 4)$. This caused the pixels near the $(5, 11)$ target to be grouped with the pixel cluster around the $(2, 2)$ target causing the estimates for both targets to be errant. Since the initial K-means cluster found centroids more than 3.25 ft away from the true target positions for both of these targets, a second iteration would not introduce an improved result. Recall that Section 2.7 describes a drawback to K-means clustering; this process does not guarantee a global optimum. How the centroid is first calculated can have a significant impact on the outcome.

In Fig. 4.13, an experiment was performed with the same target locations as Fig. 4.11. For this localization, parameters were changed such that $\Delta_p = .25$ ft, $\alpha = 150$, and

$T_c = 2\sigma_n$. The lower pixel width results in a increased amount of pixels. Additionally, a lower threshold value, $T_c = 2\sigma_n$ results in an increased amount of pixels above threshold. As seen in Fig. 4.14, there are more pixel locations to cluster than seen in Fig. 4.12. A denser cluster of pixels around the target position increases the opportunity to find a centroid among the denser crowd of pixels which are congregated around the respective targets. The smaller number of isolated pixels would have a insignificant affect on the centroid calculation. The localization error after the first K-means iteration was $\bar{e} = .52$ ft. After the second iteration, the error was $\bar{e} = .472$ ft.



(a) Target position.

(b) Image scene.

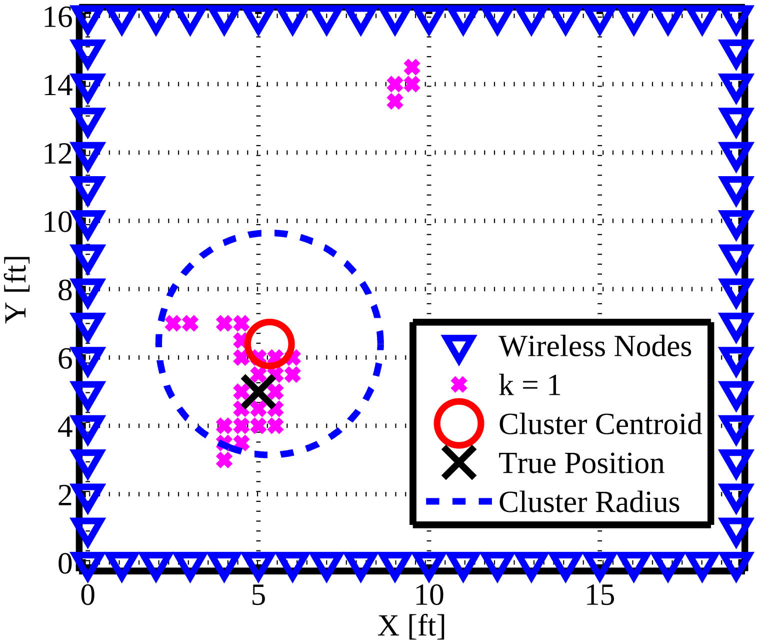
(c) Histogram of the image.

(d) Pixel locations above threshold

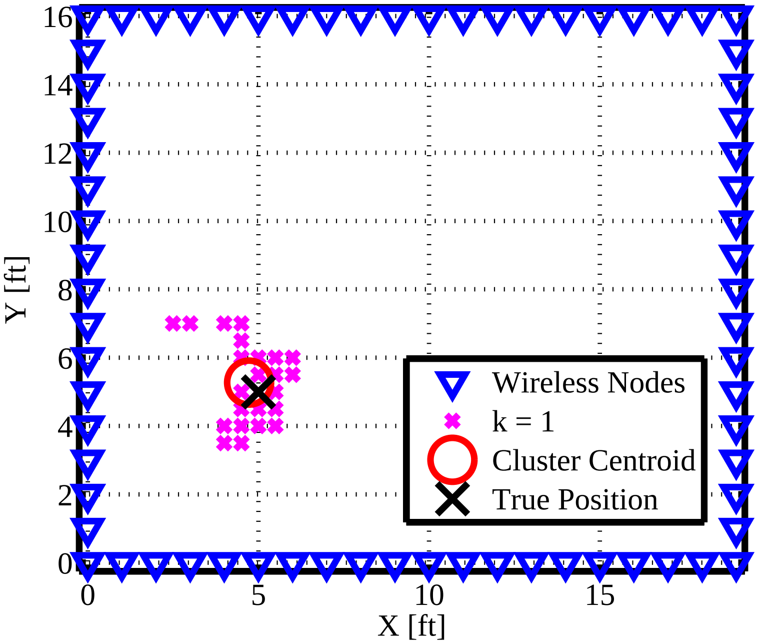
(e) Target localization after first iteration

(f) Target localization after second iteration

Figure 4.4: Localization of 1 target at (5, 5) ft with $\alpha = 250$ and $\Delta_p = 0.5$ ft.

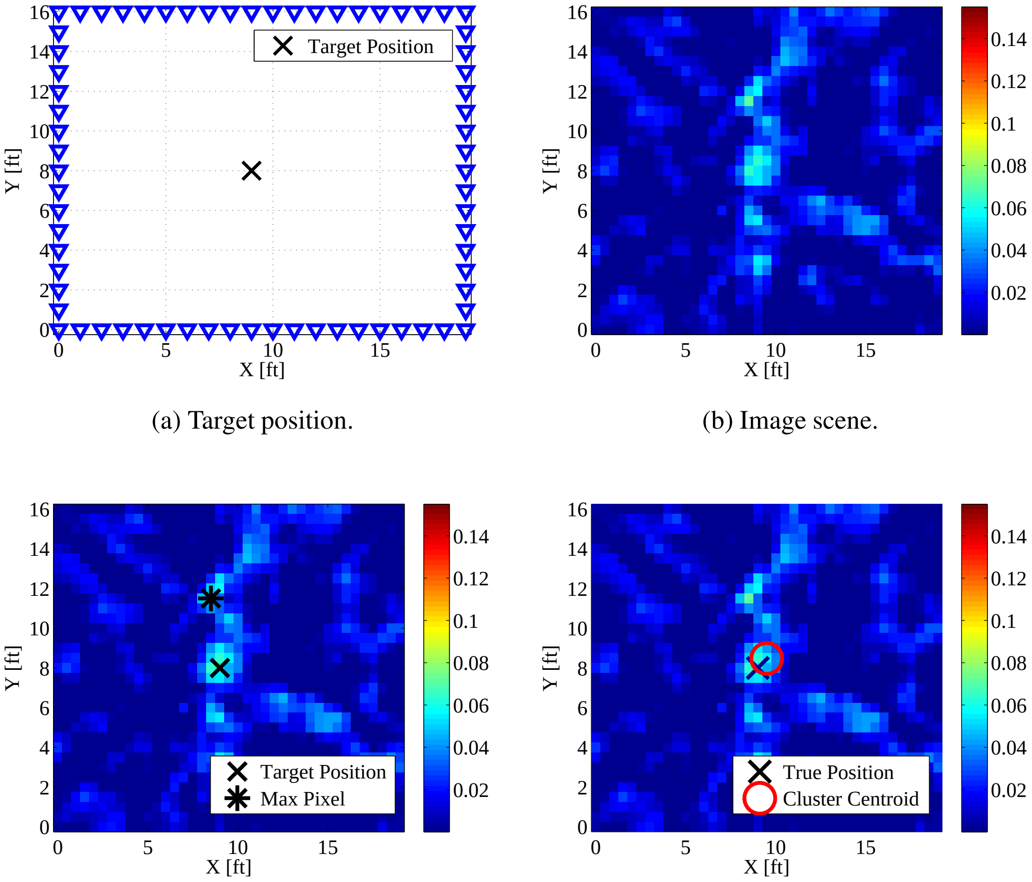


(a) First iteration of K-means clustering.



(b) Second iteration of K-means clustering.

Figure 4.5: K-means Localization: target at $(5, 5)$ ft. The RMSE after the first K-means iteration was $\bar{e} = 1.43$ ft. After the second iteration, the RMSE was $\bar{e} = 0.38$ ft.



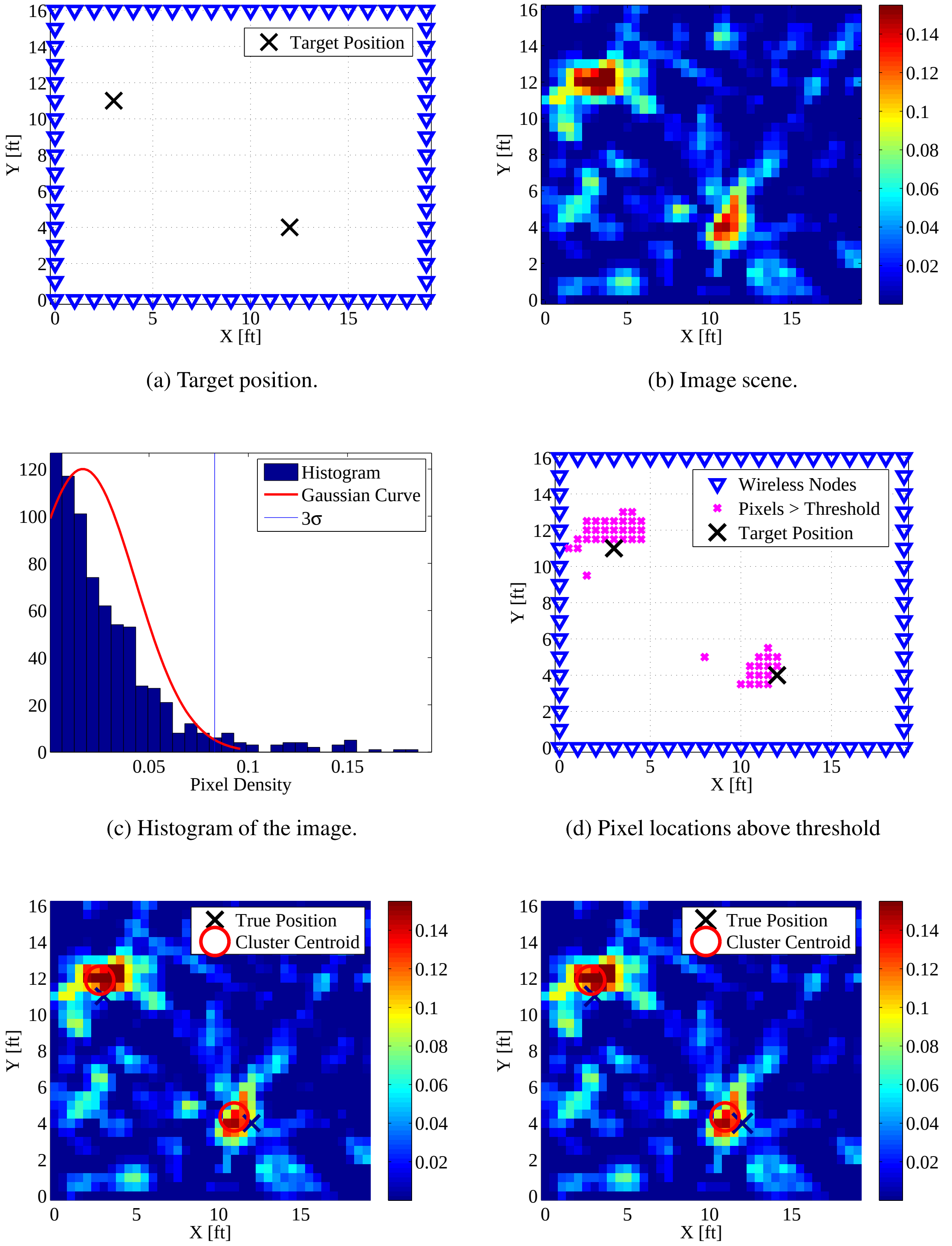
(a) Target position.

(b) Image scene.

(c) Maximum pixel intensity localization.

(d) K-means localization.

Figure 4.6: Localization of 1 target at (9, 8) ft with $\alpha = 250$, $\Delta_p = 0.5$ ft, and $T_c = 3\sigma_n$. The RMSE for the maximum pixel intensity localization was $\bar{e} = 3.54$ ft. The RMSE for the K-means clustering localization was $\bar{e} = 0.69$ ft.



(a) Target position.

(b) Image scene.

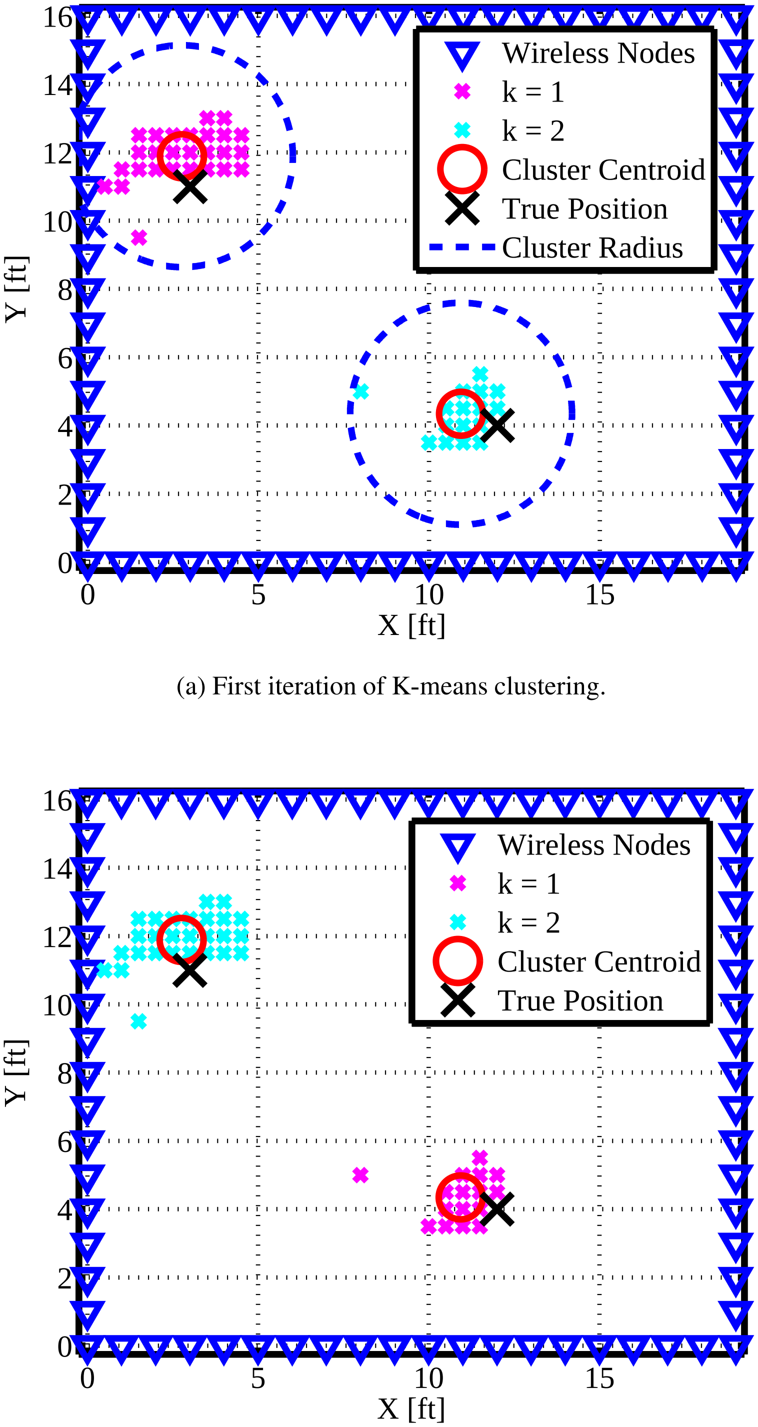
(c) Histogram of the image.

(d) Pixel locations above threshold

(e) Target localization after first iteration

(f) Target localization after second iteration

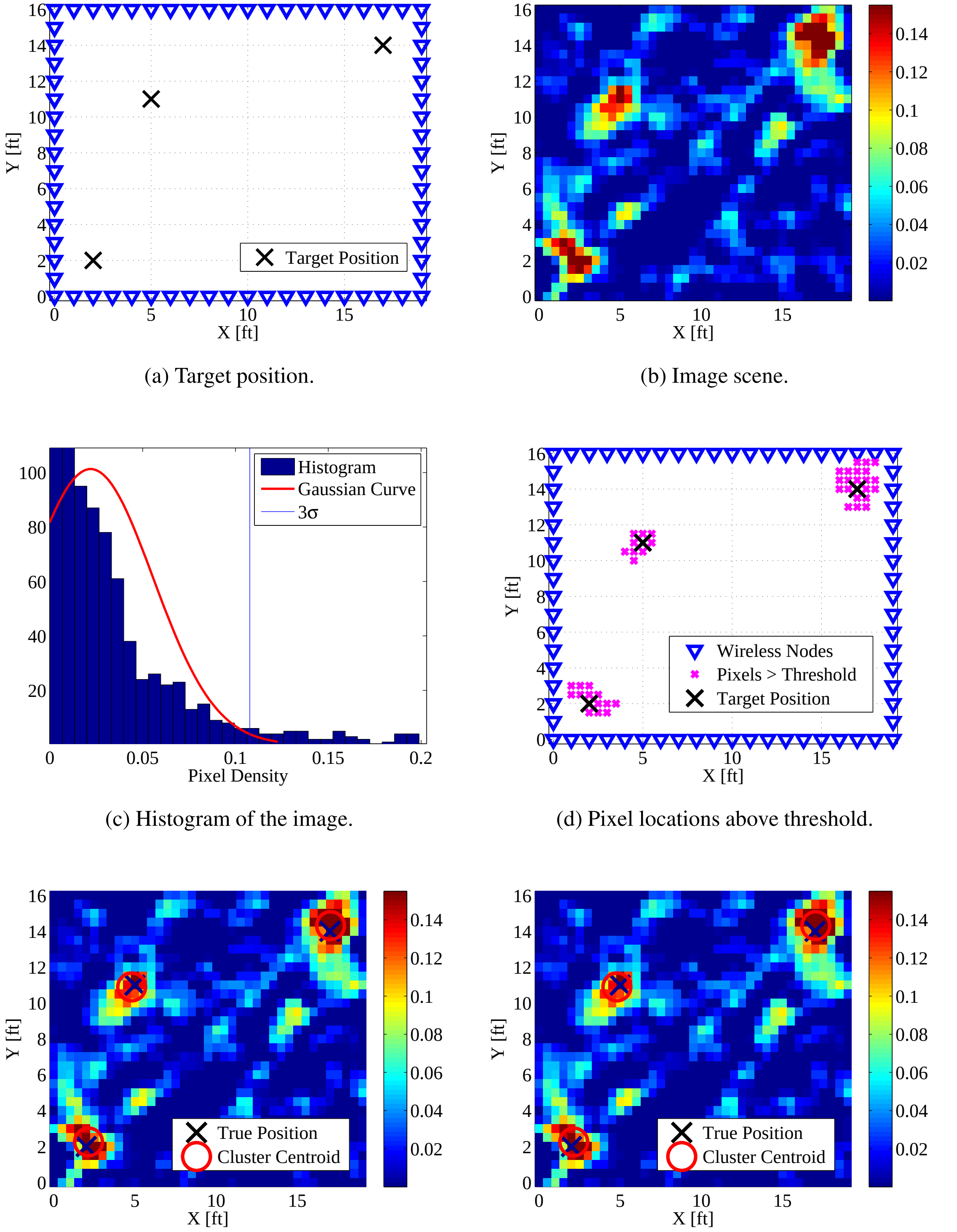
Figure 4.7: Localization of 2 targets at (3, 11) and (12, 4) ft with $\alpha = 250$ and $\Delta_p = 0.5$ ft.



(a) First iteration of K-means clustering.

(b) Second iteration of K-means clustering.

Figure 4.8: K-means Localization: 2 targets at $(3, 11)$ and $(12, 4)$ ft. The RMSE after the first K-means iteration was $\bar{e} = 0.78$ ft. After the second iteration, the RMSE was $\bar{e} = 0.78$ ft. There was no change due to the same pixels being clustered.



(a) Target position.

(b) Image scene.

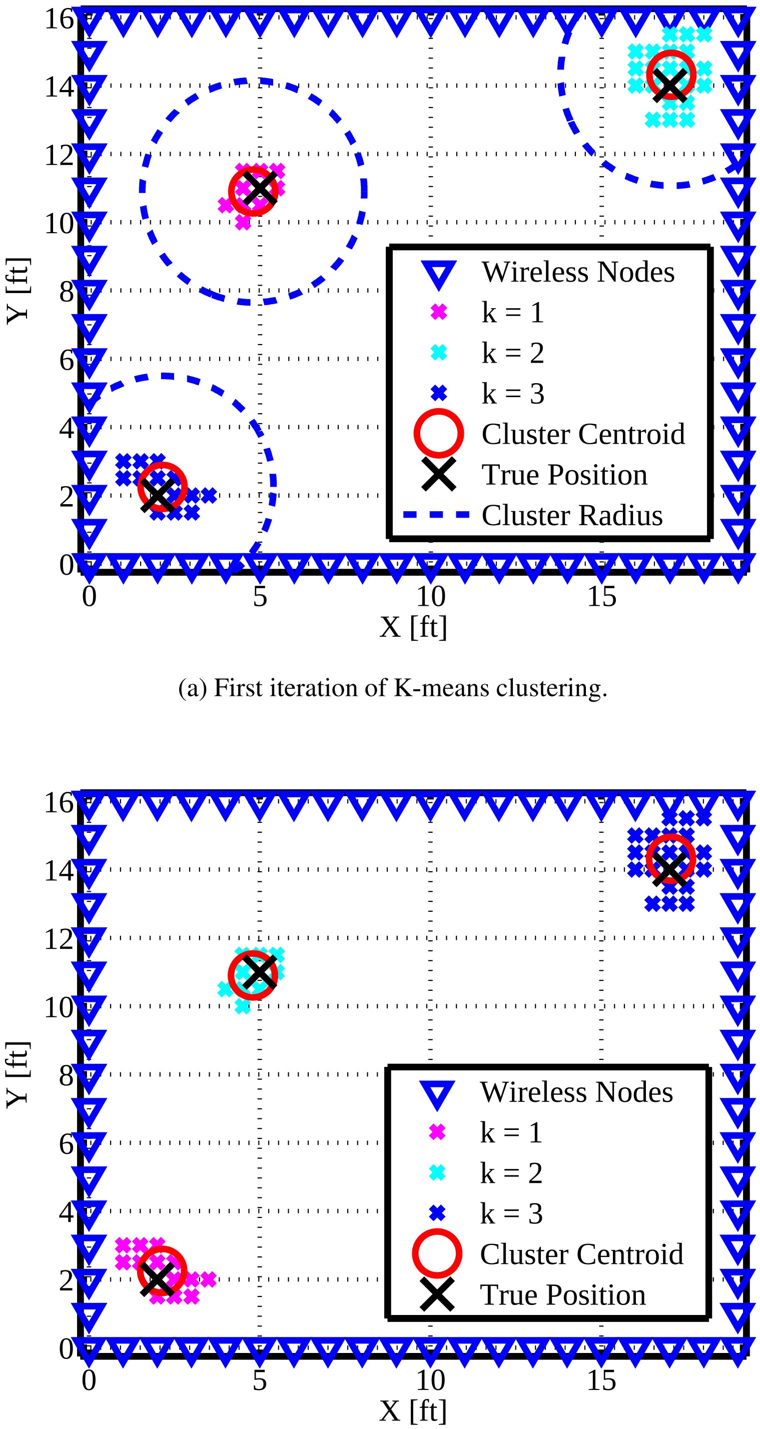
(c) Histogram of the image.

(d) Pixel locations above threshold.

(e) Target localization after first iteration.

(f) Target localization after second iteration

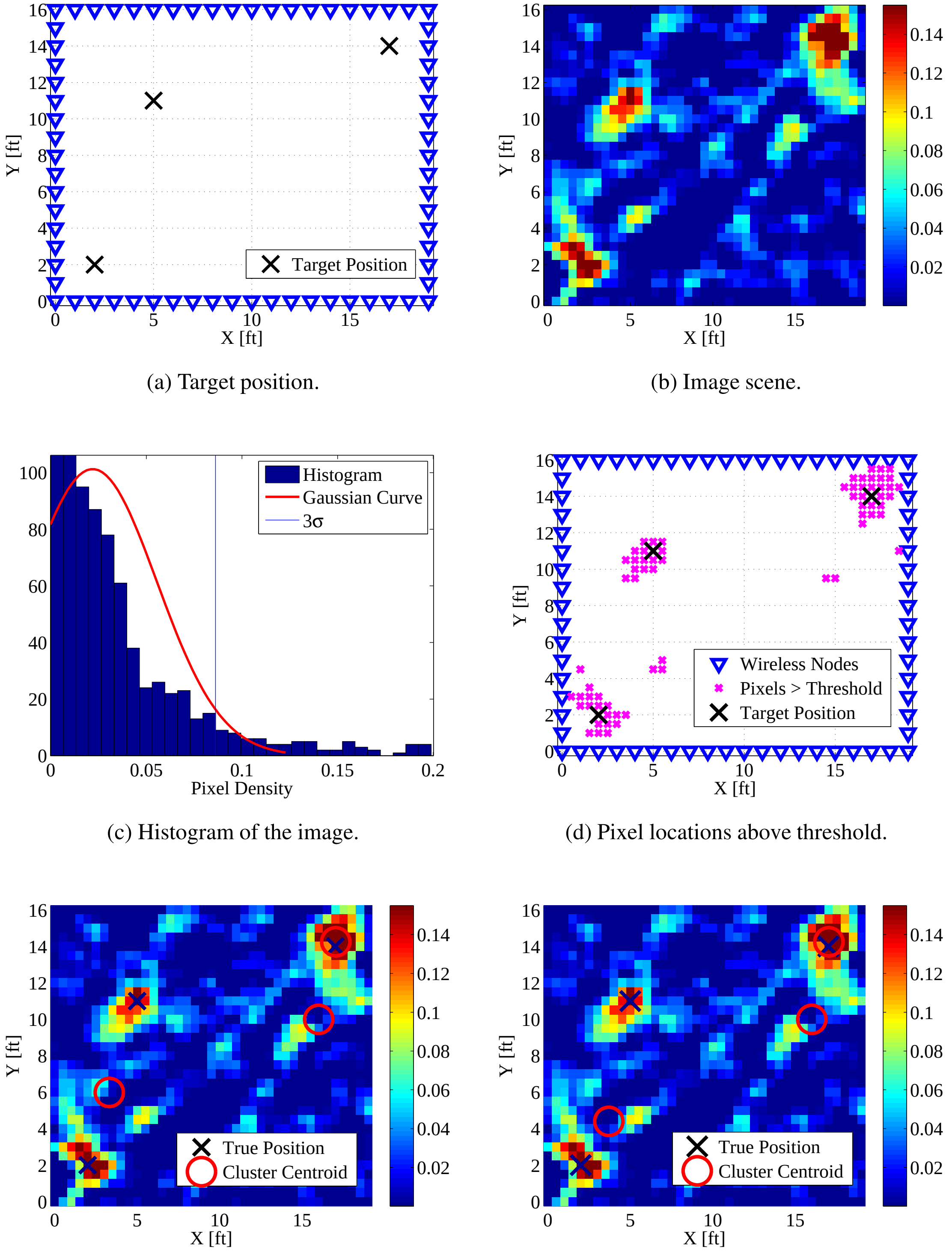
Figure 4.9: Localization of 3 targets at $(2, 2)$, $(5, 11)$, and $(17, 14)$ ft with $\alpha = 250$ and $\Delta_p = 0.5$ ft.



(a) First iteration of K-means clustering.

(b) Second iteration of K-means clustering.

Figure 4.10: K-means Localization: targets at (2, 2), (5, 11), and (17, 14) ft. The RMSE after the first K-means iteration was $\bar{e} = 0.28$ ft. After the second iteration, the RMSE was $\bar{e} = 0.28$ ft.



(a) Target position.

(b) Image scene.

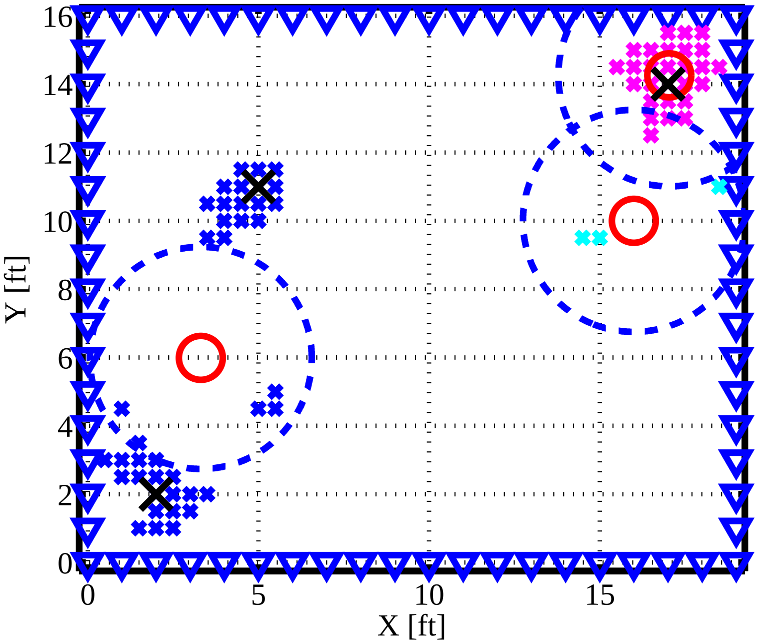
(c) Histogram of the image.

(d) Pixel locations above threshold.

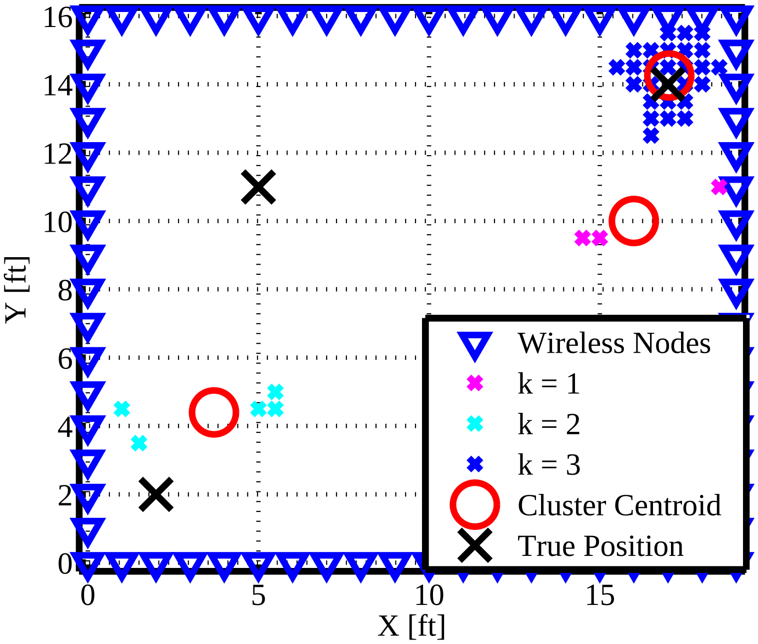
(e) Target localization after first iteration.

(f) Target localization after second iteration

Figure 4.11: Localization of 3 targets at $(2, 2)$, $(5, 11)$, and $(17, 14)$ ft with $\alpha = 250$ and $\Delta_p = 0.5$ ft.

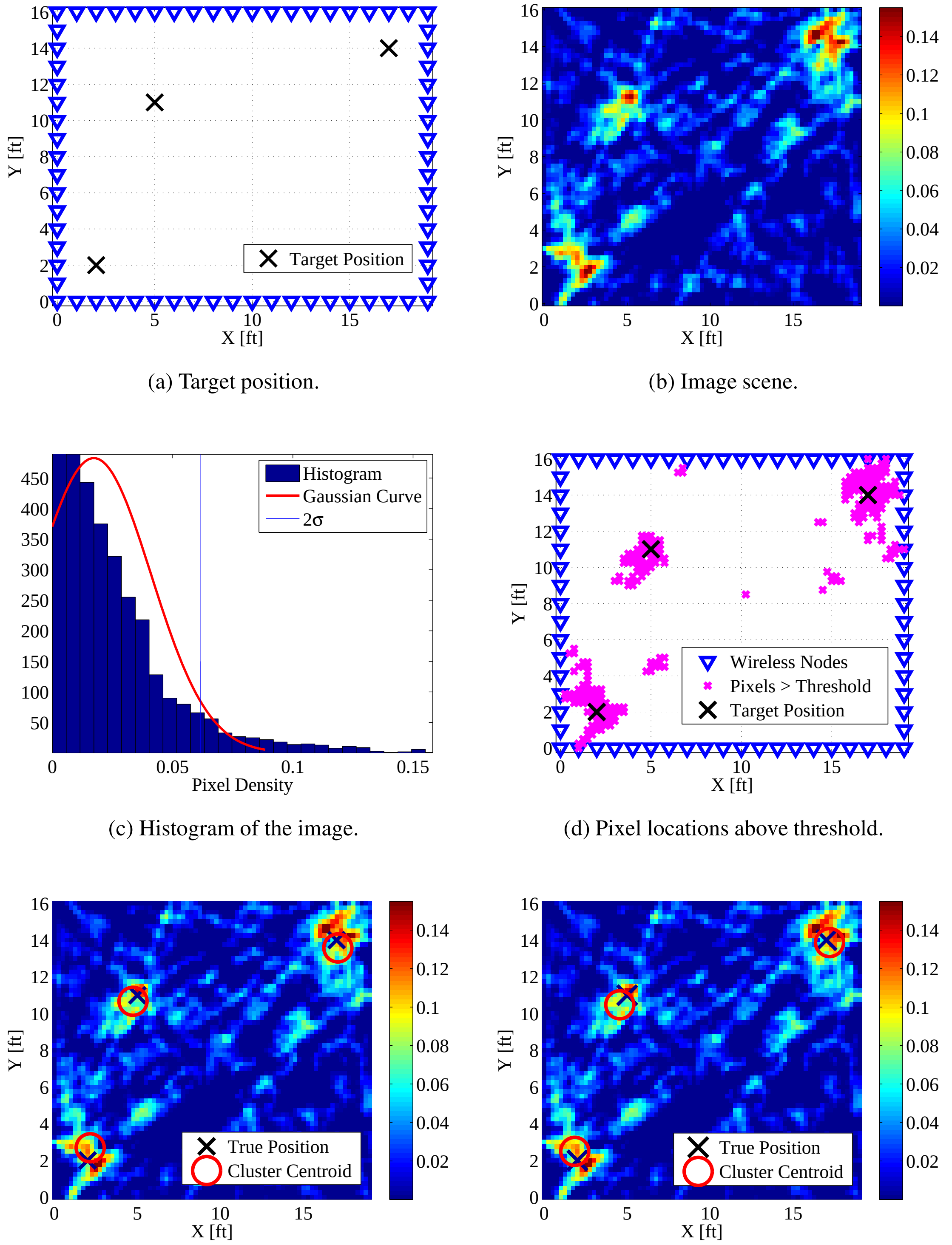


(a) First iteration of K-means clustering.



(b) Second iteration of K-means clustering.

Figure 4.12: 3 targets at (2, 2), (5, 11), and (17, 14) ft. The RMSE after the first K-means iteration was $\bar{e} = 5.17$ ft. After the second iteration, the RMSE was $\bar{e} = 4.75$ ft.



(a) Target position.

(b) Image scene.

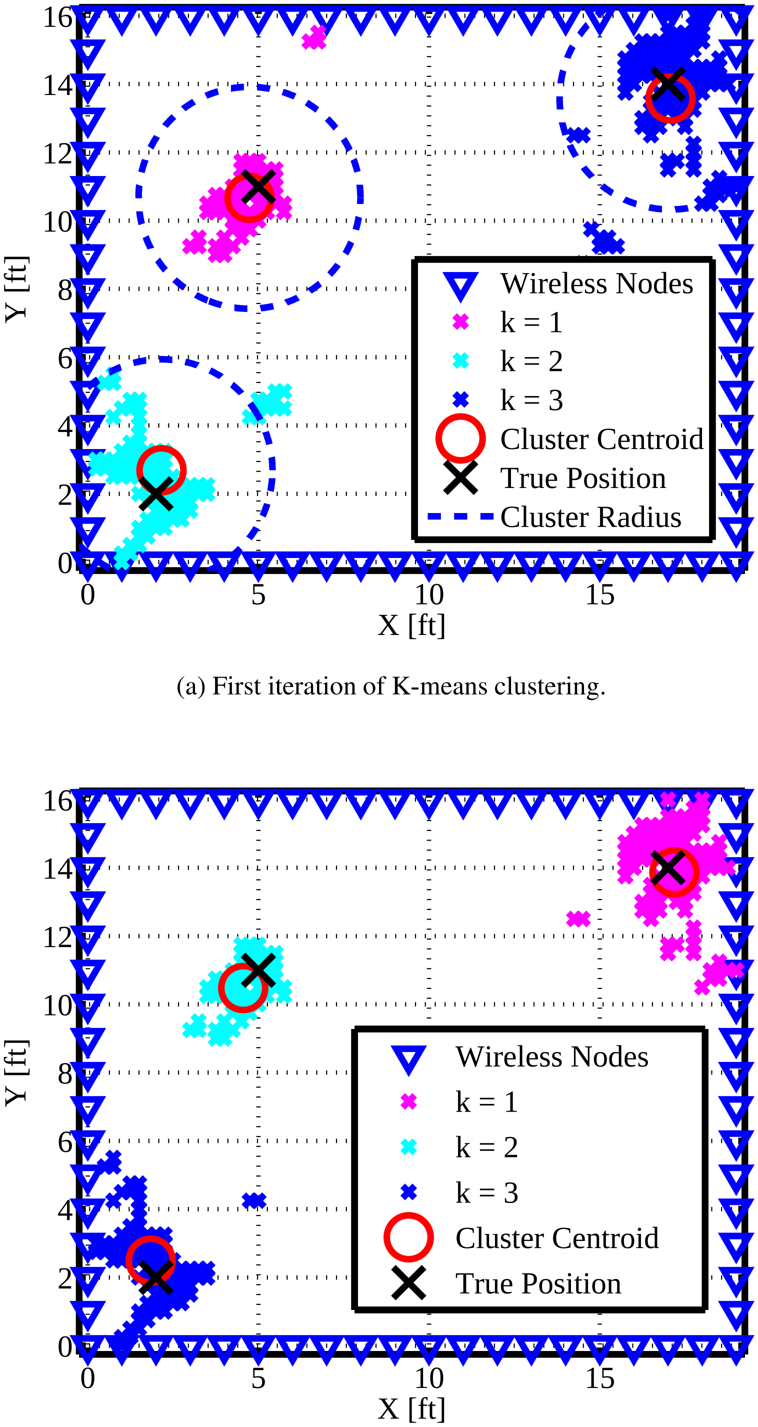
(c) Histogram of the image.

(d) Pixel locations above threshold.

(e) Target localization after first iteration.

(f) Target localization after second iteration

Figure 4.13: Localization of 3 targets at (2, 2), (5, 11), and (17, 14) ft with $\alpha = 150$, $T_c = 2\sigma_n$, and $\Delta_p = 0.25$ ft.



(a) First iteration of K-means clustering.

(b) Second iteration of K-means clustering.

Figure 4.14: 3 targets at (2, 2), (5, 11), and (17, 14) ft. The RMSE after the first K-means iteration was $\bar{e}$ = .52 ft. After the second iteration, the RMSE was $\bar{e}$ = .472 ft.

4.3 Motion Tracking

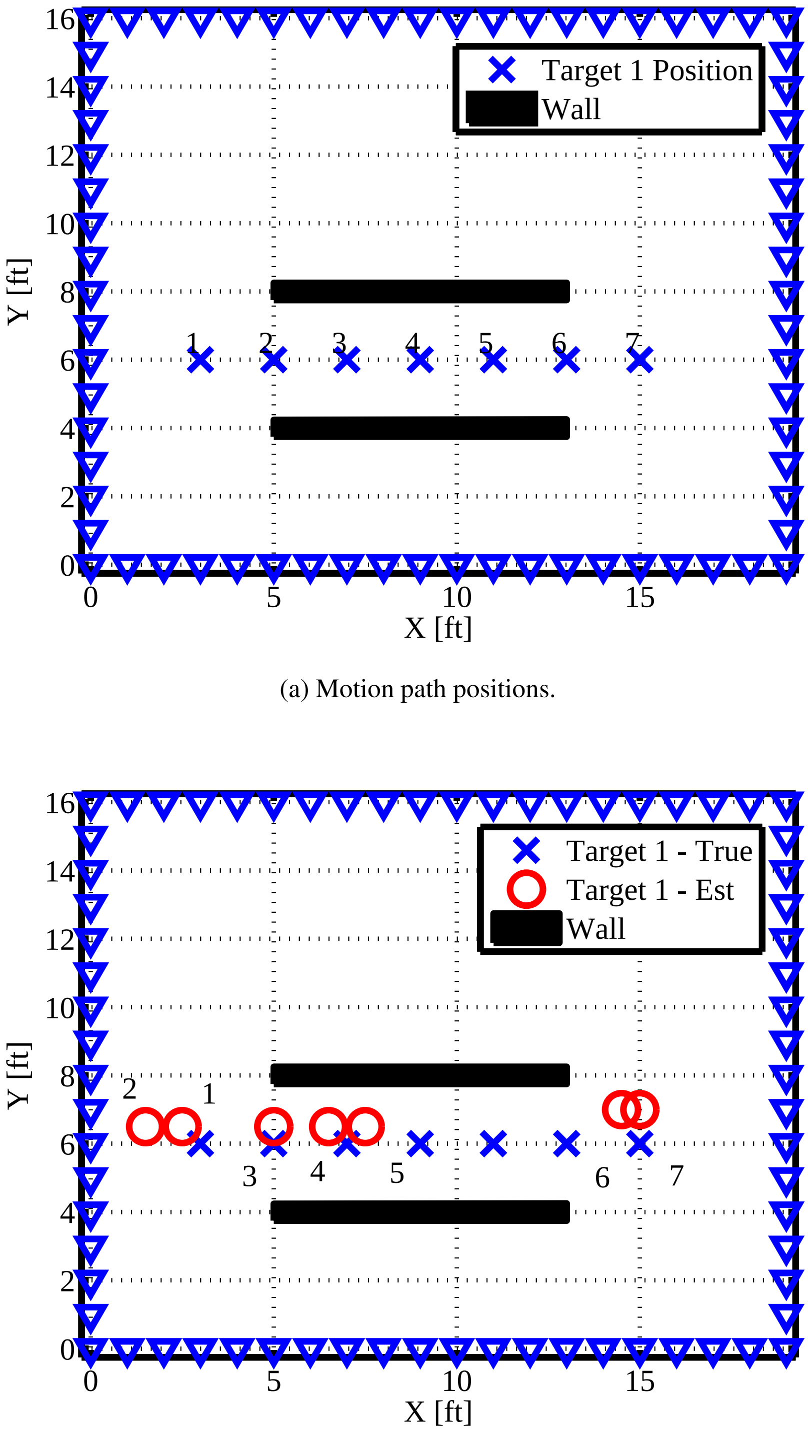
A series of motion tracking experiments were performed in this research. Since 3 target localization can be inaccurate, all motion tracking experiments were done with 1 and 2 targets. The problem with 3 targets is that it is more susceptible to mistake outliers as a target as shown in Section 4.2.2. Although localization with obstructions was not the objective of this research, obstructions were used to portray real indoor scenarios. With 1 target, K-means localization is able to be compared to the maximum pixel density estimator. For all motion tracking, 2 iterations of K-means are performed, where the cluster centroid(s) are considered the estimated target positions. For 2 targets, the goal was to analyze if they target can be successfully localized if the targets are standing close together, such as within the $R_c = 3.25$ ft radius.

4.3.1 Single Target Motion Tracking.

In Fig. 4.15, a foam wall was setup where each wall had a dimension of $[L \times W \times H] \approx [6 \times 2 \times 6]$ ft. The walls were setup to simulate a hallway with a width of approximately 4 ft. The target walked through the simulated hallway at which the target's position was estimated at each position using the maximum pixel density and K-means clustering. Although the objective of the calibration is to neutralize any obstructions inside the network, obstructions made the image noisier. Due to the image being nosier, at some frames, the maximum pixel density was further away from the true target position, but the cluster of pixels above the threshold, T_c would be closer to the target position. This resulted in a RMSE of $\bar{e} = 2.68$ ft for the maximum pixel density localization and K-means localization had a RMSE of $\bar{e} = 0.91$ ft. Fig. 4.18 illustrates a square motion tracking path for one target. Similarly, K-means localization had a lower RMSE than the maximum pixel density localization due to frames that contained maximum pixel density which were further away than the clusters of higher density pixels around the target position.

4.3.2 Two Target Motion Tracking.

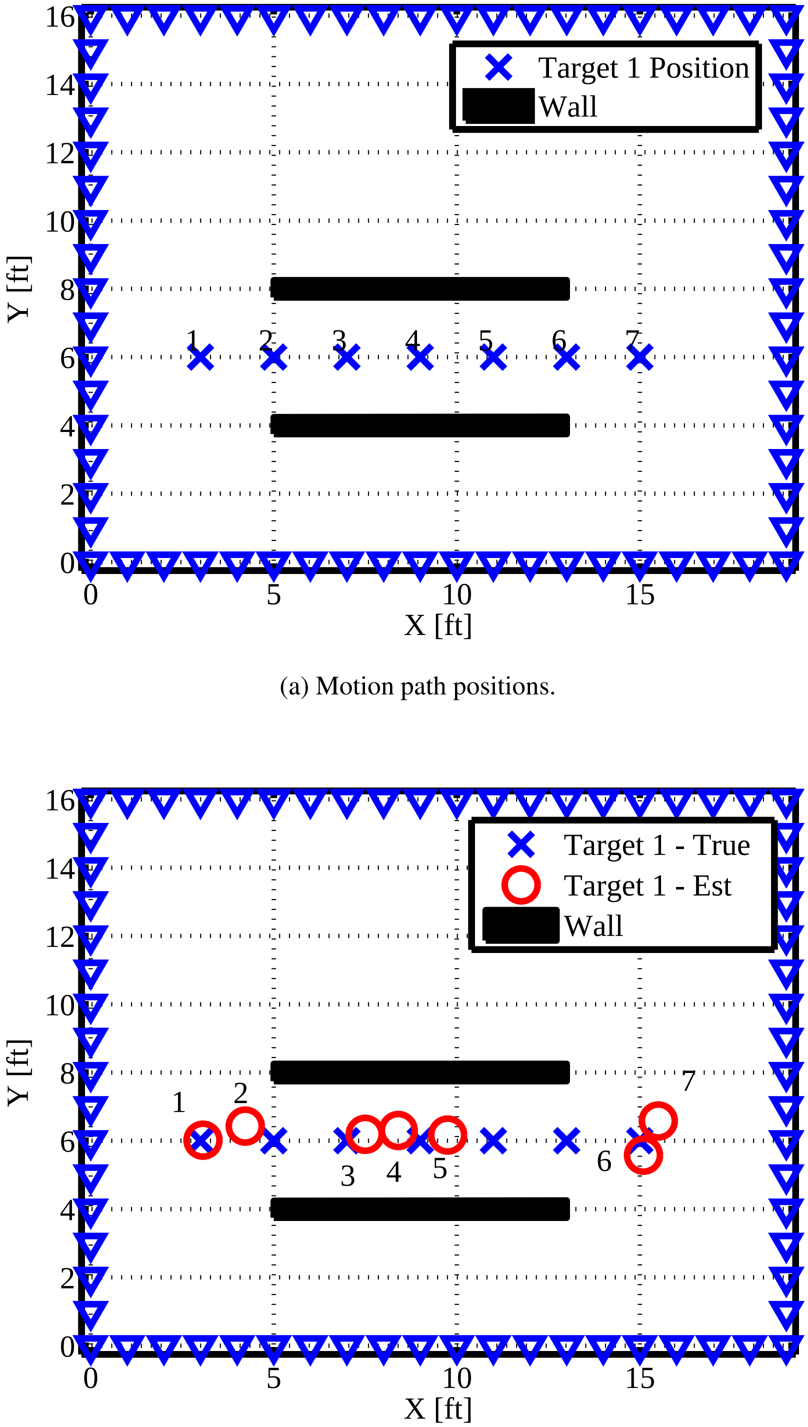
The results for motion target tracking with two targets using K-means localization are presented in Fig. 4.20. To simulate an indoor environment with obstructions, two chairs with an approximate aerial dimension of $[2 \times 2]$ ft were placed centered at $(4, 11)$ and $(14, 5)$ ft. The main objective of this motion path was to analyze the performance of K-means when the targets end up less than $R_c = 3.25$ ft away from one another. For this motion tracking path, K-means localization had a RMSE $\bar{e} = 0.59$. When the targets were within 2 ft from one another, the RMSE for that frame was less than 1 ft.



(a) Motion path positions.

(b) Localization estimate at each position.

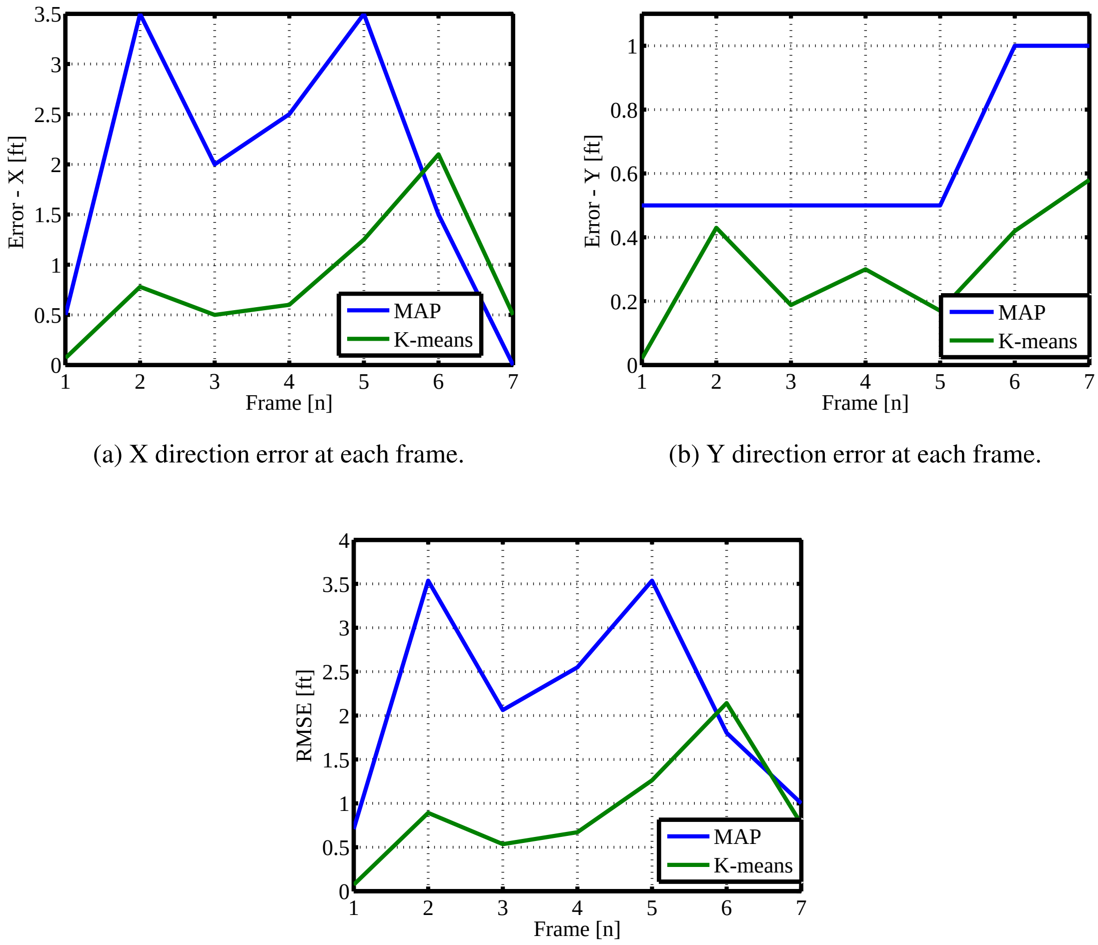
Figure 4.15: Motion tracking localization of 1 target going through a simulated hallway with walls inside the network MAP localization. The pixel with the highest density value was used to geolocate the target at each frame with $\alpha = 250$ and $\Delta_p = 0.5$ ft.



(a) Motion path positions.

(b) Localization estimate at each position.

Figure 4.16: Motion tracking localization of 1 target going through a simulated hallway with walls inside the network. K-means localization was used to geolocate the target at each frame with $T_c = 3\sigma_n$, $\alpha = 250$, and $\Delta_p = 0.5$ ft.

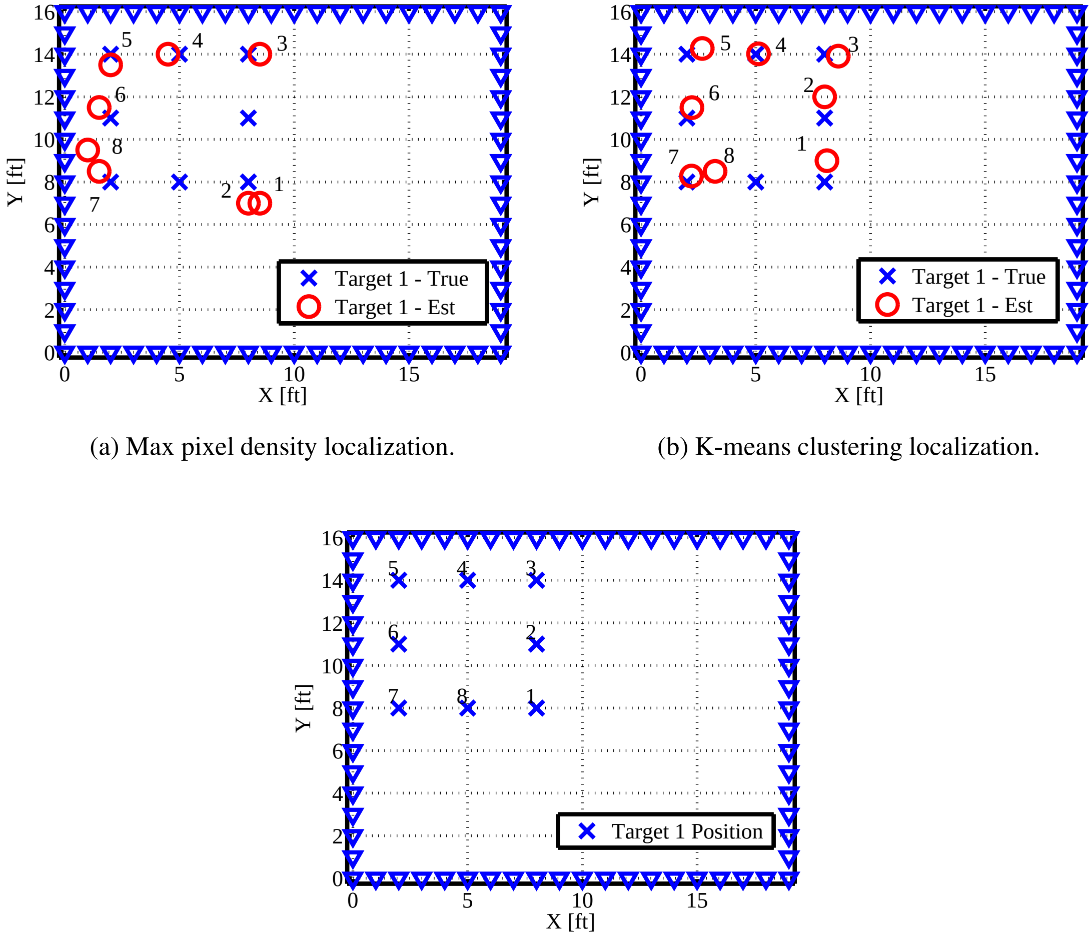


(a) X direction error at each frame.

(b) Y direction error at each frame.

(c) RMSE at each frame.

Figure 4.17: The $\overline{RMSE}$ for the target over the motion tracking path through the walls was $\bar{e} = 2.68$ ft using the maximum pixel density estimate. The $\overline{RMSE}$ for the target over the motion tracking path through the walls was $\bar{e} = 0.91$ ft using K-means localization.

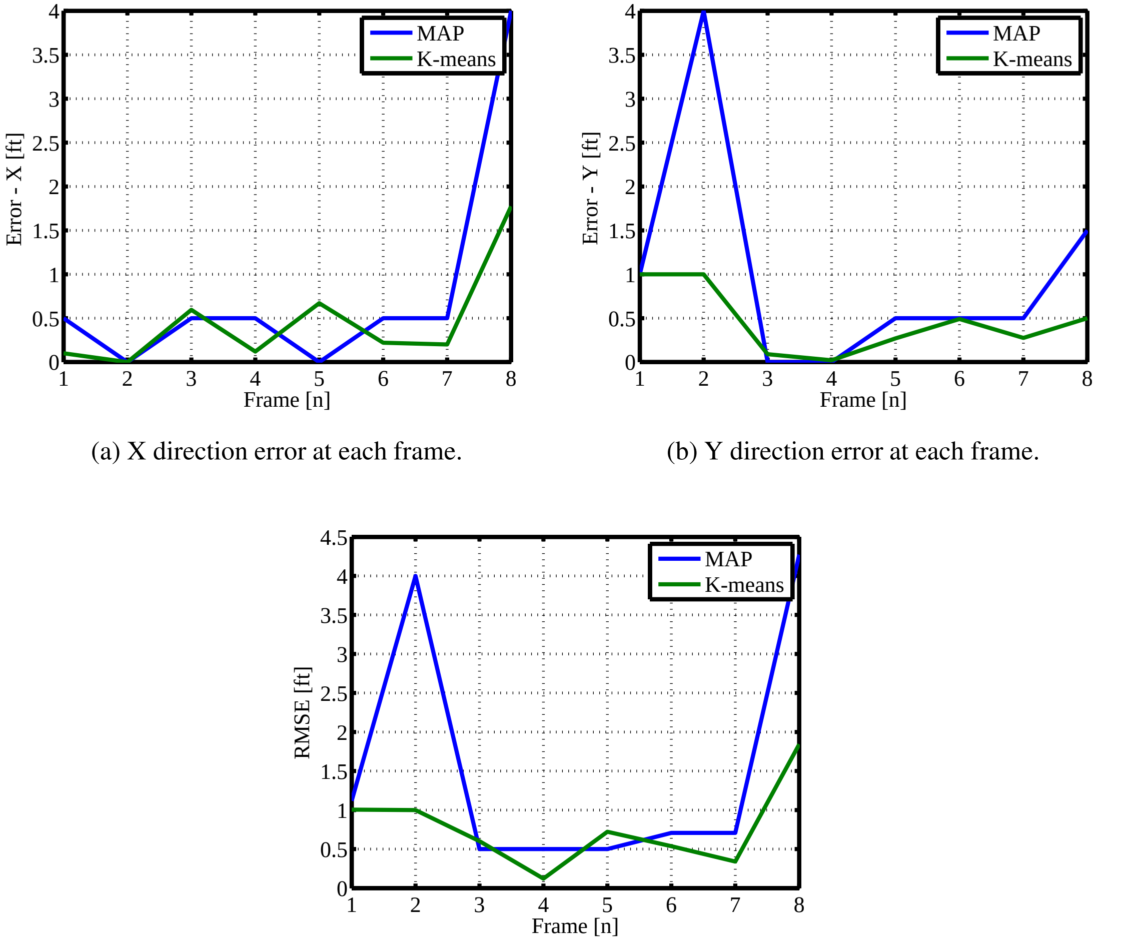


(a) Max pixel density localization.

(b) K-means clustering localization.

(c) True position of target at each frame.

Figure 4.18: A single target moves throughout the network in a square starting at $(8,8)$ and ending at $(5,8)$ ft. Maximum pixel density and K-means clustering are both used to localize the target position for comparison.

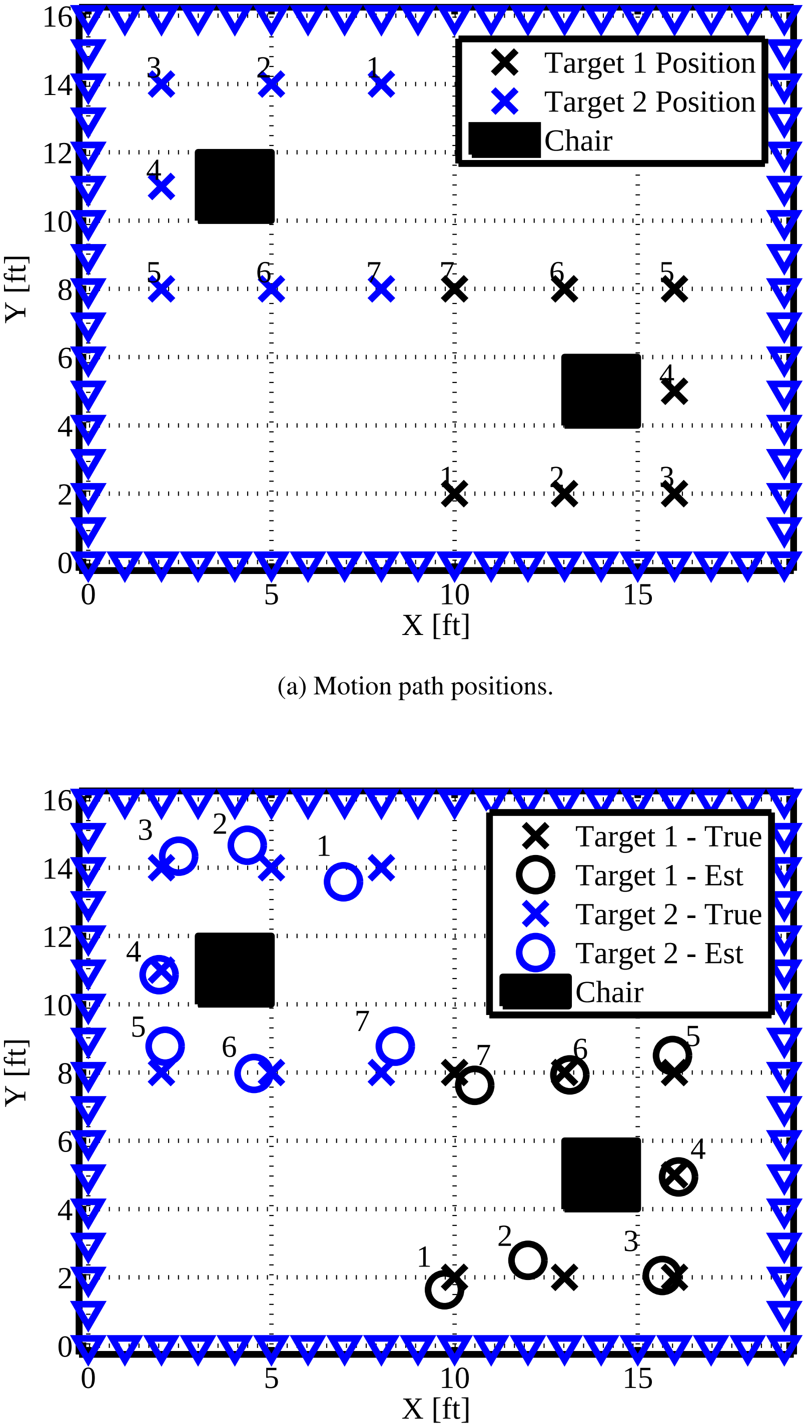


(a) X direction error at each frame.

(b) Y direction error at each frame.

(c) RMSE at each frame.

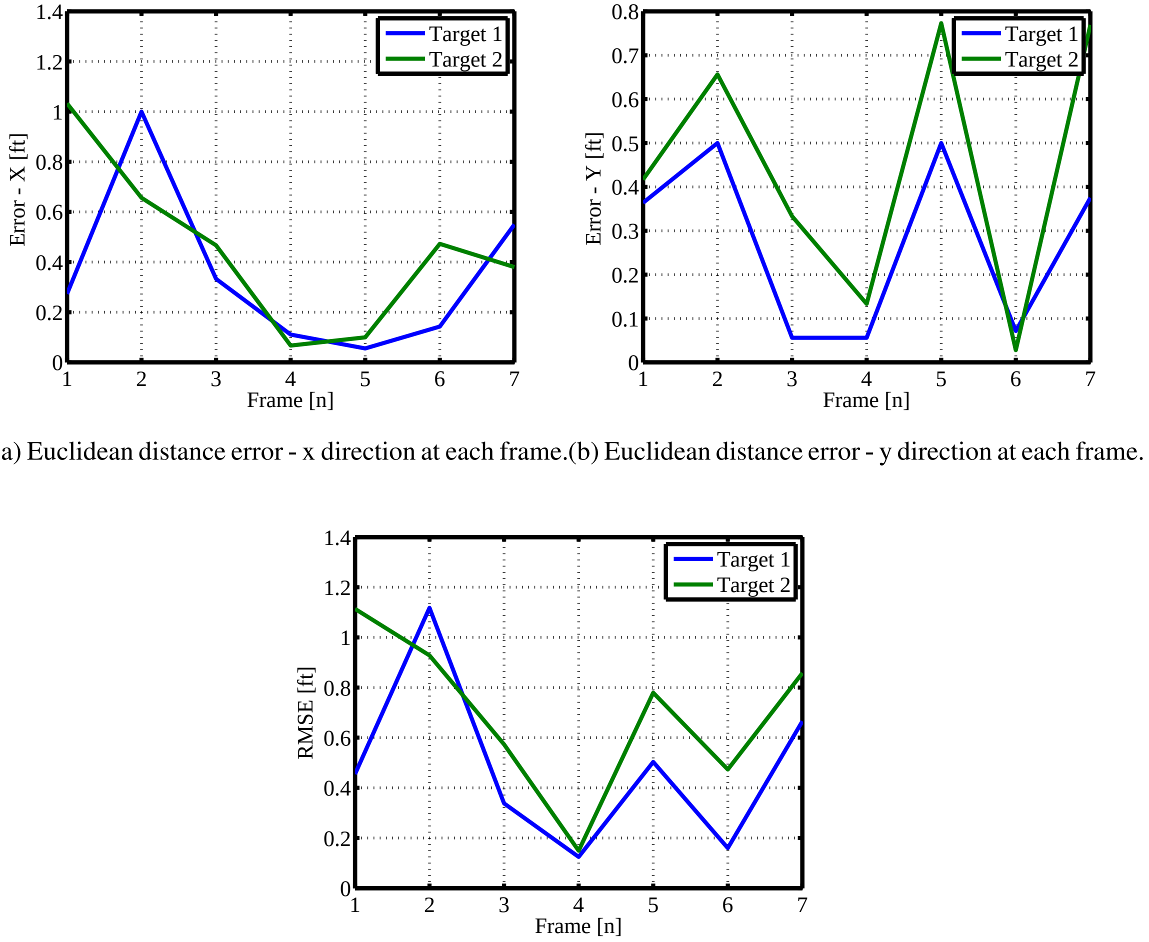
Figure 4.19: The $\overline{RMSE}$ for the target over the motion tracking path from Fig. 4.18 using maximum pixel density localization was $\bar{e} = 1.54$ ft. The $\overline{RMSE}$ for the target over the motion tracking path from Fig. 4.18 using K-means clustering localization was $\bar{e} = 0.77$ ft.



(a) Motion path positions.

(b) Localization estimates at each position.

Figure 4.20: Motion tracking localization of 2 targets with obstructions. K-means clustering was used to geolocate the targets at each frame with $T_c = 3\sigma_n$, $\alpha = 250$, and $\Delta_p = 0.5$ ft.



(a) Euclidean distance error - x direction at each frame.(b) Euclidean distance error - y direction at each frame.

(c) RMSE at each frame.

Figure 4.21: The $\overline{RMSE}$ for the two targets over the motion tracking path with obstructions was $\bar{e} = .59$ ft.

4.4 Chapter Summary

This chapter reviewed the results of various stationary localization with one, two, and three targets. With 1 target localization, K-means clustering was able to be compared to maximum pixel density localization to show in which situation K-means could be more robust in estimating the target location. Localization for two targets was more consistent than localization with three targets. Motion tracking experiments were performed in a variety of different simulated situations to analyze the performance of K-means localization. The choice of pixel threshold and number of pixels above the threshold play a key role in determining which pixels of interest will be clustered together to localize the targets inside the network.

V. Conclusion and Future Work

THIS chapter summarizes the methodology, results, and conclusions made from this thesis as well as provides a recommendation for future work. With the growing research interest of RTI, single target localization has been the primary focus. This research was motivated to explore multiple target localization for situations where localizing more than one target would be beneficial. Additionally, this research had the desire to look at a robust means of clustering together pixels of higher densities as opposed to only using the pixel with the highest density.

This thesis explored a new means to localize multiple targets using a pixel threshold to localize pixels above a certain pixel density. The methods in which the image frames were estimated were presented in Chapter 2. The use of the weighting model, regularization, and image estimate have been explored and commonly used in RTI research. Practical applications with movement, obstructions, through the wall, and outdoor environments have been explored by the research community. The challenge has been estimating the position of more than one target [2].

K-means clustering is a known algorithm used in other data mining applications such as among machine learning applications, pattern recognition, hyper-spectral imagery, artificial intelligence, crowd analysis, and MTT [11], [12], [13]. However, the method in how it is applied is new to the RTI research community. The objective of the pixel density threshold is to be robust enough, that as the statistics of each frame change with the number of targets and where the targets are in the network, the threshold would segregate pixels of higher densities. These pixels would relate to where the targets are inside the network. Additionally, with a radius to ignore pixels outside a set distance from the initial centroids, possible errant pixel densities could be ignored. Since the computational cost of K-means is relatively low as discussed in Chapter 2, two iterations of K-means could potentially produce more accurate localization estimates than one.

In a series of stationary localization experiments, this research was able to analyze the performance of K-means for one, two, and three targets in an indoor network. For one target localization, if the maximum pixel density was further away from the target position than the cluster of pixels above the threshold were, K-means localization was more accurate. K-means performed equally as accurate as single target localization. However, if there were no pixels outside the cluster radius, a second iteration of K-means did not produce any change in results as the centroid locations would understandably stay the same. Three target localization was found to be inaccurate. Changing the pixel width, Δ_p to be lower, which increased the number of pixels in the network and lowering the pixel threshold, T_c provided a larger amount of pixels to be clustered, which increased the performance of K-means. A higher amount of pixels were found to aid K-means clustering in finding a solution that minimized the inter cluster variance.

For motion tracking images, the image scene estimates were found to be noisier. Thus, for both single target tracking situations K-means performed more accurately than using the highest pixel density to localize the target. For two targets, K-means was able to localize two targets that moved towards one another. When the targets were approximately 2 ft from each other, the RMSE at that frame was less than 1 ft. Due to three target localization not being as accurate and having a network limited in size, three target localization for motion tracking was not performed.

This research showed that K-means can be applied to one or more targets in a RTI network. Further work is recommended to improve the process and make it more robust for multiple targets. The future work section has recommendations on future research areas that can expand on this research and other similar research areas of RTI.

5.1 Future Work

Automatic Target Recognition. For K-means to be successful, the number of targets needs to be known. In this research, it is assumed the number of targets are known prior to estimating the locations of the targets. Implementing a method to estimate the number of targets can be beneficial in applications where the number of targets may not be known [11], [12].

K-Medoids Clustering. K-means does not guarantee to return a global optimum. The final solution is sensitive to the initial set of clusters [13]. Although computationally more expensive, using K-medoids could potentially be a viable solution for MTT with RTI. K-medoids could potentially be more robust due to minimizing the sum of general pairwise dissimilarities, which could minimize the negative effects of noise and outliers [57]. The trade off between performance and computational complexity could be examined.

Adaptive Filter Tracking. This research did not use any adaptive filters to track moving targets. The use of adaptive filters such as the Kalman filter or Gaussian particle filter could minimize the RMSE. The use of adaptive filers would minimize the effects of observation noise or other variables that could lead to inaccurate position estimates [10], [11], [12], [58].

Bibliography

[1] J. Wilson and N. Patwari, "Radio Tomographic Imaging with Wireless Networks," *Transactions on Mobile Computing*, vol. 9, no. 5, pp. 621–632, Jan. 2010.

[2] Y. Zhao, N. Patwari, and J. M. Phillips, "Radio Tomographic Imaging and Tracking of Stationary and Moving People via Kernel Distance," in *Proc. of the 12th International Conference of Information Processing in Sensor Networks*. ACM, Apr. 2013, pp. 229–240.

[3] Memsic Technologies, "TelosB TPR2420 Mote Platform Datasheet," Webpage, http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf, 2013.

[4] C. Huang and Y. Tseng, "The Coverage Problem in a Wireless Sensor Network," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, Aug. 2005.

[5] N. Patwari, J.N. Ash, S. Kyperountas, Spyros, A.O. Hero III, R.L. Moses, and N.S. Correal, "Locating the Nodes: Cooperative Localization in Wireless Sensor Networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[6] J. Wilson and N. Patwari, "Through-Wall Tracking Using Variance-Based Radio Tomography Networks," *arXiv preprint arXiv:0909.5417*, Sep. 2009.

[7] T. Pham, B.M. Sadler, and H. Papadopoulos, "Energy-Based Source Localization via Ad-hoc Acoustic Sensor Network," in *Proc. IEEE Workshop on Statistical Signal Processing*, Oct. 2003, pp. 387–390.

[8] M. Bocca, O. Kaltiokallio, N. Patwari, and S. Venkatasubramanian, "Multiple Target Tracking with RF Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1787–1800, Jul. 2014.

[9] S. Nannuru, Y. Li, M. Coates, and B. Yang, "Multi-Target Device-Free Tracking using Radio Frequency Tomography," in *Proc. Seventh Int'l Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Dec. 2011, pp. 508–513.

[10] M. McCracken, M. Bocca, and N. Patwari, "Joint Ultra-wideband and Signal Strength-based Through-building Tracking for Tactical Operations," in *IEEE Internation Conference on Sensing, Communications and Netowkring*, Jun. 2013, pp. 309–317.

[11] N. Patwari M. Bocca, O. Kalktiokallio and S. Venkatasubramanian, "Multiple Target Tracking with RF Sensor Networks," *IEEE Trasnaction on Mobile Computing*, vol. 1, no. 1, pp. 1–12, Feb. 2013.

[12] M. Anderson, J. Rydell, L. St-Laurent, D. Prevost, and F. Gustafsson, "Crowd Analysis with Target Tracking, K-Means Clustering and Hidden Markhov Models," in *Information Fusion, 2012 15th International Conference*, Jul. 2012, pp. 1903–1910.

[13] C. Zhang and S. Xia, "K-means Clustering Algorithm with Improved Initial Center," in *Second International Workshop on Knowledge Discovery and data Mining, 2009*. IEEE, Jan. 2009, pp. 790–792.

[14] W. C. J. Hunter, H. H. Barrett, and L. R. Furenlid, "Calibration Method for ML Estimation of 3D Interaction Position in a Thick Gamma-Ray Detector," *IEEE Transactions on Nuclear Science*, vol. 56, no. 1, pp. 189–196, Feb. 2009.

[15] R. Niu and V. K. Pramod, "Target Location Estimation in Sensor Networks With Quantized Data," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4519–4528, Dec. 2006.

[16] G. Sun, J. Chen, W. Guo, and K. J. R. Liu, "Joint TDOA and AOA Location Algorithm," *Journal of Systems Engineering and Electronics*, vol. 24, no. 2, pp. 183–188, Apr. 2013.

[17] B. Sobhani, E. Paolini, A. Giorgetti, M. Mazzotti, and M. Chiani, "Target Tracking for UWB Multistatic Radar Sensor Networks," *Journal of Selected Topics in Signal Processing: Special Issue on Non-Cooperative Localization Networks*, vol. 8, no. 1, Feb. 2014.

[18] M. Chiani, A. Giorgetti, M. Mazzotti, R. Minutolo, and E. Paolini, "Target Detection Metrics and Tracking for UWB Radar Sensor Networks," in *IEEE International Conference on Ultra-Wideband, 2009*, Sep. 2009, pp. 469–474.

[19] Y. Kilic, H. Wymeersch, A. Meijerink, M.J. Bentum, and W.G. Scanlon, "Device-Free Person Detection and Ranging in UWB Networks," *Journal of Selected Topics in Signal Processing: Special Issue on Non-Cooperative Localization Networks*, vol. 8, no. 1, May 2013.

[20] R.S. Blum A. M. Haimovich and R.S. Blum, "MIMO Radar with Widely Separated Antennas," *Signal Processing Magazine, IEEE*, vol. 25, no. 1, pp. 116–129, Dec. 2007.

[21] R. Sundaram, R.K. Martin, and C. Anderson, "Regularization in Radio Tomographic Imaging," in *Proc. Eighth Conf. on Wireless Sensing, Localization, and Processing*. International Society for Optics and Photonics, May 2013, vol. 8753.

[22] N. Pirzada, M. Y. Nayan, F. Subhan. M. F. Hassan, and M. A. Khan, "Device-Free Localization Technique for Indoor Detection and Tracking of Human Body: A Survery," in *Internation Conference on Innovation, Management and Technology Research*. ScienceDirect, Sep. 2013, pp. 422–429.

[23] O. Kaltiokallio, M. Bocca, and N. Patwari, "A Fade Level-Based Spatial Model for Radio Tomographic Imaging," *IEEE Transaction on Mobile Computing*, vol. 13, no. 6, pp. 1159–1172, Dec. 2013.

[24] H. Hashemi, "The Indoor Radio Propagation Channel," *Proc. of the IEEE*, vol. 81, no. 7, pp. 943–968, Jul. 1993.

[25] M. Kanso and M.G. Rabbat, "Efficient Detection and Localization of Assets in Emergency Situations," *Proc. Thirty-Seventh Int'l Sym. Medical Info. Commun. Tech.*, vol. 1, pp. 1, Feb. 2009.

[26] M.A. Kanso and M.G. Rabbat, "Compressed RF Tomography for Wireless Sensor Networks: Centralized and Decentralized Approaches," in *Proc. Fifth Int'l Conf. on Distributed Computing in Sensor Systems (DCOSS)*, pp. 173–186. Springer, Jun. 2009.

[27] Y. Li, X. Chen, M. Coates, and B. Yang, "Sequential Monte Carlo Radio-Frequency Tomographic Tracking," in *Proc. Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2011, pp. 3976–3979.

[28] B.R. Hamilton, X. Ma, B.J. Baxley, and S.M. Matechik, "Propagation Modeling for Radio Frequency Tomography in Wireless Networks," *Journal of Selected Topics in Signal Processing: Special Issue on Non-Cooperative Localization Networks*, vol. 8, no. 1, Oct. 2013.

[29] R. Ganesh and K. Pahlavan, "Effects of Traffic and Local Movements on Multipath Characteristics of an Indoor Radio Channel," *IET Electronics Letters*, vol. 26, no. 6, pp. 810–812, Aug. 2002.

[30] A. Folkerts R. K. Martin and T. Heinl, "Accuracy vs. Resolution in Radio Tomography," *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 2480–2491, May 2014.

[31] C.R. Anderson, R.K. Martin, T.O. Walker, and R.W. Thomas, "Radio Tomography for Roadside Surveillance," *Journal of Selected Topics in Signal Processing: Special Issue on Non-Cooperative Localization Networks*, vol. 8, no. 1, Feb. 2014.

[32] J. Wilson, N. Patwari, and F.G. Vasquez, "Regularization Methods for Radio Tomographic Imaging," in *Proc. Virginia Tech Symposium on Wireless Personal Communications*, Jun. 2009.

[33] Y. Mostofi and A. Gonzalez-Ruiz, "Compressive Cooperative Obstacle Mapping in Mobile Networks," in *Proc. Conf. on Military Communications (MILCOM)*, Nov. 2010, pp. 524–530.

[34] B.R. Hamilton, "Applications of Bayesian Filtering in Wireless Networks: Clock Synchronization, Localization, and RF Tomography," *Ph.D. Dissertation*, 2012.

[35] M. Bocca, A. Luong, N. Patwari, and T. Schmid, "Dial It In: Rotating RF Sensors to Enhance Radio Tomography," Jul. 2013, pp. 600–608.

[36] N. Patwari and P. Agrawal, "Effects of Correlated Shadowing: Connectivity, Localization, and RF Tomography," in *Proc. Int'l Conf. on Information Processing in Sensor Networks (IPSN)*, Apr. 2008, pp. 82–93.

[37] Y. Zhao and N. Patwari, "Robust Estimators for Variance-Based Device-Free Localization and Tracking," *IEEE Transaction on Mobile Computing*, vol. PP, no. 99, pp. 1–1, Dec. 2014.

[38] J. Wilson and N. Patwari, "See-Through Walls: Motion Tracking using Variance-Based Radio Tomography Networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 612–621, Sep. 2011.

[39] Y. Zhao and N. Patwari, "Noise Reduction for Variance-Based Device-Free Localization and Tracking," in *Proc. Eighth Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Jun. 2011, pp. 179–187.

[40] O. Kaltiokallio, M. Bocca, and N. Patwari, "Enhancing the Accuracy of Radio Tomographic Imaging using Channel Diversity," in *Proc. IEEE Ninth Int'l Conf. on Mobile Adhoc and Sensor Systems (MASS)*, Oct. 2012, pp. 254–262.

[41] Y. Mostofi, "Compressive cooperative sensing and mapping in mobile networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 12, pp. 1769–1784, Oct. 2011.

[42] R.K. Martin, C. Anderson, R.W. Thomas, and A.S. King, "Modelling and Analysis of Radio Tomography," in *Proc. Fourth Int'l Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, Dec. 2011, pp. 377–380.

[43] G. Oliveri D. Trinchero F. Viani, P. Rocca and A. Massa, "Localization, Tracking, and Imaging of Targets in Wireless Sensor Networks: An Invited Review," *Radio Science*, vol. 46, no. 5, pp. RS5002, 2011.

[44] P. Agrawal and N. Patwari, "Correlated Link Shadow Fading in Multi-hop Wireless Networks," *Wireless Communications*, vol. 8, no. 8, pp. 4024–4036, Aug. 2009.

[45] J. Wilson and N. Patwari, "A Fade-Level Skew-Laplace Signal Strength Model for Device-Free Localization with Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 947–958, Jun. 2012.

[46] N. Patwari and P. Agrawal, "NeSh: A Joint Shadowing Model for Links in a Multi-hop Network," in *Proc. Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Apr. 2008, pp. 2873–2876.

[47] Y. Zhao and N. Patwari, "Demo Abstract: Histogram Distance-Based Radio Tomographic Localization," in *ACM/IEEE 11th International Conference on Information Processin in Sensor Networks*, Apr. 2012, pp. 129–130.

[48] A. W. Hu B. Wei, A. Varshney, T. Voigt N. Patwari, and C. T. Chou, "dRTI: Directional Radio Tomographic Imaging," *arXiv preprint arXiv:1402.2744*, vol. 1, no. 1, Feb. 2014.

[49] M. Perkins N. S. Correal N. Patwari, A. O. Hero and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[50] F. K. P. Shareena, "An Efficient Approac for Multi-Target Tracking in Sensor Networks Using Ant Colony Opitmization," *International Journal for Scientific Research and Development*, vol. 1, no. 7, pp. 2321–0613, Oct. 2013.

[51] G. Mao, B. Fidan, and B. Anderson, "Wireless Sensor Network Localization Techniques," *Computer networks*, vol. 51, no. 10, pp. 2529–2553, Jul. 2007.

[52] Y. Selkowitz et all C. Fylor, C. Vinschen, "Cygwin Project Home Page," Webpage, http://www.cygwin.com/, 2013.

[53] Stanford Research Community, "TinyOS Home Page," Webpage, http://www.tinyos.net, 2007-13.

[54] Neal Patwari; University of Utah, "SPAN Home Page," Webpage, http://span.ece.utah.edu, 2013.

[55] Texas Instruments, "CC2420 Inverted F Antenna Design Note DN0007," Webpage, http://www.ti.com/lit/an/swru120b/swru120b.pdf, 2008.

[56] R.K. Martin, A. Folkerts, and T. Heinl, "Accuracy vs. Resolution in Radio Tomography," Mar. 2014, vol. 62, pp. 2480–2491.

[57] B. Pardeshi and D. Toshniwal, "Improved K-Medoids Clustering Based on Cluster Validity Index and Object Density," in *2010 IEEE 2nd International Advance Computing Conferece (IACC)*, Feb. 2010, pp. 379–384.

[58] J. H. Kotecha and P. M. Djuric, "Gaussian Particle Filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 26–03–2015	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Aug 2013–Mar 2015

4. TITLE AND SUBTITLE

Estimating Single and Multiple Target Locations Using K-Means Clustering With Radio Tomographic Imaging In Wireless Sensor Networks

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Nishida, Jeffrey K., Captain, USAF

5d. PROJECT NUMBER

15G451I

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-15-M-038

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR (Attn: Dr. Tristan Nguyen, AFOSR/RTC)
Comm. Phone: (703) 696-7796 / DSN: 426-7796 / Fax: (703) 696-7360
Email: tristan.nguyen@afosr.af.mil
875 North Randolph Street, Suite 325, Room 3112, Arlington, VA 22203-1768

10. SPONSOR/MONITOR'S ACRONYM(S)

Air Force office of Scientific Research (AFOSR)/RTC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Geolocation involves using data from a sensor network to assess and estimate the location of a moving or stationary target. RSS, AoA, and/or TDoA measurements can be used to estimate target location in sensor networks. RTI is an emerging DFL concept that utilizes the RSS values of a WSN to geolocate stationary or moving target(s). The WSN is set up around the AoI and the target of interest, which can be a person or object. The target inside the AoI creates a shadowing loss between each link being obstructed by the target. This research focuses on position estimation of single and multiple targets inside a RTI network. This research applies K-means clustering to localize one or more targets. K-means clustering is an algorithm that has been used in data mining applications such as machine learning applications, pattern recognition, hyper-spectral imagery, artificial intelligence, crowd analysis, and MTT.

15. SUBJECT TERMS

Radio Tomographic Imaging, RTI, wireless sensor network, localization, RSS, DFL, WSN, clustering, MTT

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Richard K. Martin (ENG)
					19b. TELEPHONE NUMBER *(include area code)*
U	U	U	UU	109	(937) 255-3636 x4625 richard.martin@afit.edu